



Universitat d'Alacant  
Universidad de Alicante

Scene Understanding for  
Mobile Robots exploiting  
Deep Learning Techniques

José Carlos Rangel Ortiz



Tesis

**Doctorales**

[www.eltallerdigital.com](http://www.eltallerdigital.com)

UNIVERSIDAD de ALICANTE



Universitat d'Alacant  
Universidad de Alicante

Instituto Universitario de Investigación Informática

# Scene Understanding for Mobile Robots exploiting Deep Learning Techniques

José Carlos Rangel Ortiz

TESIS PRESENTADA PARA ASPIRAR AL GRADO DE  
DOCTOR POR LA UNIVERSIDAD DE ALICANTE

MENCIÓN DE DOCTOR INTERNACIONAL

PROGRAMA DE DOCTORADO EN INFORMÁTICA

Dirigida por:

Dr. Miguel Ángel Cazorla Quevedo

Dr. Jesús Martínez Gómez

2017



The thesis presented in this document has been reviewed and approved  
for the  
INTERNATIONAL PhD HONOURABLE MENTION

I would like to thank the advises and contributions for this thesis of the  
external reviewers:

Professor Eduardo Mario Nebot  
(University of Sydney)

Professor Cristian Iván Pinzón Trejos  
(Universidad Tecnológica de Panamá)



This thesis is licensed under a CC BY-NC-SA International License (Creative Commons AttributionNonCommercial-ShareAlike 4.0 International License). You are free to share — copy and redistribute the material in any medium or format Adapt — remix, transform, and build upon the material. The licensor cannot revoke these freedoms as long as you follow the license terms. Under the following terms: Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. NonCommercial — You may not use the material for commercial purposes. ShareAlike — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.

Document by José Carlos Rangel Ortiz.

---

Universitat d'Alacant  
Universidad de Alicante





*A mis padres.  
A Tango, Tomás y Julia  
en su memoria.*



Universitat d'Alacant  
Universidad de Alicante





*Revuelvo la mirada y a veces siento espanto  
cuando no veo el camino que a ti me ha de tornar...*

*¡Quizá nunca supiese que te quería tanto,  
si el Hado no dispone que atravesara el mar!...*

*Patria (fragmento)*

**Ricardo Miró**



Universitat d'Alacant  
Universidad de Alicante



# Agradecimientos

---

Al culminar esta tesis doctoral y mirar hace casi 4 años cuando inicié, me percaté de lo mucho que ha ocurrido y agradezco profundamente la fortuna de encontrar personas maravillosas dispuestas a brindarme lo mejor de sí en pos de mi mejora como investigador y como persona, y con los cuales estaré eternamente agradecido.

Dentro de estas personas debo agradecer a mis directores, Miguel y Jesús, por la paciencia, las charlas, por siempre tener un momento, por aclararme las cosas repetidamente cada vez que el código se iba por donde no debía y por compartir sus conocimientos y experiencias tanto de la vida como de la profesión. Muchas gracias a ambos por llevar este reto conmigo y por siempre tener esa palabra de aliento para seguir adelante.

Al grupo de investigación RoVit y al Departamento de Ciencia de la Computación e Inteligencia Artificial. A los chicos del laboratorio Fran, Sergio, Félix, Ed, Vicente y Manolo por esas charlas sobre series y sus buenos consejos para avanzar. A los miembros del DTIC que siempre me vieron como uno más, por todas esas comidas y momentos que me han hecho olvidar la lejanía de mi patria.

A Cristina e Ismael en la UCLM en Albacete por abrirme las puertas de su grupo y compartirme sus conocimientos. Al grupo de *Data Intelligence* de la Universidad Jean Monnet de Saint-Etienne en Francia, donde tuve la oportunidad de realizar mi estancia. A Marc y Elissa por su guía y consejos. A María, Leo y Dennis con quienes compartí tantos buenos momentos. *Merci Beaucoup!* De igual manera a la Escuela de Doctorado de la Universidad de Alicante la cual aportó los recursos para realizar esta estancia.

No puedo expresar con palabras mi agradecimiento a mis compañeros de piso, Edgardo y Ed, mi familia adoptiva; quienes han tenido que aguantar a este saxofonista aficionado que los fines de semana inundaba el piso con sus a veces desafinadas notas, por esos cines, cenas y vacaciones y viajes. Muchas gracias por permitirme cambiar de tema y olvidar las frustraciones de un *paper*. A mis amigos en Santiago para los cuales la lejanía nunca fue excusa para mantener el contacto.

A tres personas quienes me vieron iniciar este camino, mas tuvieron la oportunidad de llegar al final junto a mí, mis abuelos Tango y Tomás, y mi Tía Julia. Gracias por sus charlas, por sus cuidados y por ese ejemplo de vida y de siempre seguir adelante.

A mis familiares en Panamá por su apoyo incondicional y en especial a ti Quenda, que con tus actos me has enseñado a nunca perder la sonrisa y la esperanza de un mejor porvenir.

A mis padres que siempre me han inculcado el avanzar y que desde mis más remotos recuerdos han apoyado mis decisiones sin importar sus dudas o lo raro que les parecieran. Sin ustedes cada letra de este documento no sería posible. A mi hermano y cuñada que a pesar de la distancia siempre estuvieron dispuestos a brindarme su ayuda ante cualquier cosa que necesitase. A esa pequeña persona que estuvo allí en cada vídeo llamada, José Andrés, gracias por hacerme reír y permitirme verte crecer.

A los profesores de la Facultad de Ingeniería de Sistemas y Computación de la Universidad Tecnológica de Panamá y en especial a los del Centro Regional de Veraguas donde estudié y quienes siempre me animaron a mejorar. Finalmente deseo agradecer a la Universidad Tecnológica de Panamá y el IFARHU, las instituciones que depositaron en mí su confianza para llevar a cabo estos estudios.

Al servicio de traducción de la Universidad de Alicante quién ha revisado que las ideas expresadas en este documento cumplan las reglas que exige el idioma en el cual se redactan.

Alicante, 8 de junio de 2017

José Carlos Rangel Ortiz

# Resumen

---

Cada día los robots se están volviendo más comunes en la sociedad. En consecuencia, estos deben poseer ciertas habilidades básicas para interactuar con los seres humanos y el medio ambiente. Una de estas habilidades es la capacidad de entender los lugares por los cuales tiene la capacidad de movilizarse. La visión por computador es una de las técnicas comúnmente utilizadas para lograr este propósito. Las tecnologías actuales en este campo ofrecen soluciones sobresalientes aplicadas a datos que cada día tienen una mejor calidad permitiendo así producir resultados más precisos en el análisis de un entorno.

Con esto en mente, el objetivo principal de esta investigación es desarrollar y validar un método eficiente de comprensión de escenas basado en la presencia de objetos. Este método será capaz de ayudar en la resolución de problemas relacionados con la identificación de escenas aplicada a la robótica móvil.

Buscamos analizar los métodos más avanzados del estado-del-arte con el fin de encontrar el más adecuado según nuestros objetivos, así como también seleccionar el tipo de datos más conveniente para tratar este problema.

Otro objetivo de la investigación es determinar el tipo de datos más adecuado como fuente de información para analizar la escenas, con el fin de encontrar una representación precisa de estas mediante el uso de etiquetas semánticas o descriptores de características para nubes de puntos.

Como objetivo secundario mostraremos la bondad de la utilización de descriptores semánticos generados con modelos pre-entrenados en procesos de mapeado y clasificación de escenas, así como también el uso de modelos

de *deep learning* junto con procedimientos de descripción de características 3D para construir un modelo de clasificación de objetos 3D, lo cual está directamente relacionado con el objetivo de representación de esta investigación.

La investigación descrita en esta tesis fue motivada por la necesidad de un sistema robusto capaz de comprender las ubicaciones en las cuales un robot interactúa normalmente. De la misma manera, el advenimiento de mejores recursos computacionales ha permitido implementar algunas técnicas que fueron definidas en el pasado, las cuales demandan una alta capacidad computacional y que ofrecen una posible solución para abordar los problemas de comprensión de la escenas.

Una de estas técnicas son las Redes Neuronales Convolucionales (CNN, por sus siglas en inglés). Estas redes poseen la capacidad de clasificar una imagen basada en su apariencia visual. A continuación, generan una lista de etiquetas léxicas y la probabilidad para cada una de estas etiqueta, la cual representa cómo de probable es la presencia de un objeto en la escena. Estas etiquetas se derivan de los conjuntos de entrenamiento en los cuales las redes neuronales fueron entrenadas para reconocer. Por lo tanto, esta lista de etiquetas y probabilidades podría ser utilizada como una representación eficiente del entorno y a partir de estas asignar una categoría semántica a las locaciones donde un robot móvil es capaz de navegar, permitiendo al mismo tiempo construir un mapa semántico o topológico basado en esta representación semántica del lugar.

Después de analizar el estado-del-arte en la comprensión de escenas, hemos identificado un conjunto de enfoques con el fin de desarrollar un procedimiento robusto de comprensión de escenas. Entre estos enfoques existe una brecha casi inexplorada en lo relativo a la comprensión de escenas basadas en la presencia de objetos en estas. Consecuentemente, proponemos llevar a cabo un estudio experimental basados en este enfoque, destinado a encontrar una forma de describir plenamente una escena teniendo en cuenta los objetos que se encuentran en el lugar.

Dado que la tarea de comprensión de escenas implica la detección y anotación de objetos, uno de los primeros pasos será determinar el tipo de datos que se emplearán como datos de entrada para nuestra propuesta.

Con esto en mente, nuestra propuesta considera la evaluación del uso de datos 3D. Este tipo de datos cuenta con el inconveniente de la presencia de ruido en sus representaciones, por lo tanto, proponemos utilizar el algoritmo *Growing Neural Gas* (GNG, por sus siglas en inglés) para reducir el efecto de ruido en procedimientos de reconocimiento de objetos utilizando datos 3D. La GNG tiene la capacidad de crecer adaptando su topología para representar información 2D, produciendo una representación de menor tamaño de los datos de entrada, con una ligera influencia de ruido. Aplicado a datos 3D, la GNG presenta un buen enfoque con capacidad para afrontar los problemas derivados de la presencia de ruido en una escena.

Sin embargo, la utilización de datos 3D supone un conjunto de problemas tales como la falta de un conjunto de datos de objetos 3D con una suficiente cantidad de modelos para lograr generalizar los métodos a situaciones reales, así como el hecho de que el procesamiento de datos tridimensionales es computacionalmente costoso y requiere un enorme espacio de almacenamiento. Estos problemas nos orientaron a explorar nuevos enfoques para desarrollar tareas de reconocimiento de objetos. Por lo tanto, considerando los sobresalientes resultados obtenidos por las CNNs en las últimas ediciones del reto ImageNet, para clasificación de imágenes; proponemos llevar a cabo una evaluación de las CNNs como un sistema de detección de objetos. Estas redes fueron propuestas inicialmente desde la década de 1990 y son hoy en día de fácil implementación debido a la mejora de *hardware* acontecidas en los últimos años. Las CNN han mostrado resultados satisfactorios cuando han sido evaluadas en problemas tales como: la detección de objetos, peatones, señales de tráfico, clasificación de ondas sonoras, y para el procesamiento de imágenes médicas entre otros.

Además, un valor agregado de las CNNs es la capacidad de descripción semántica derivada de las categorías/etiquetas que la red es capaz de identificar y la cual podría traducirse como una explicación semántica de la imagen de entrada.

En consecuencia, proponemos la evaluación del uso de estas etiquetas semánticas como un descriptor de escenas, con el objetivo de construir un modelo supervisado de clasificación de escenas. Dicho esto, también proponemos el uso de este descriptor semántico para la generación de mapas



topológicos y evaluar las capacidades descriptivas de las etiquetas léxicas.

Igualmente, este descriptor semántico podría ser adecuado para el etiquetado no supervisado de un entorno, por lo que proponemos su uso en este tipo de problemas con el fin de lograr un método robusto de etiquetado de escenas.

Finalmente, con el objetivo de abordar el problema de reconocimiento de objetos proponemos desarrollar un estudio experimental para el etiquetado no supervisado de objetos. Esto será aplicado a los objetos presentes en una nube de puntos y se etiquetarán empleando una herramienta de etiquetado léxico. A continuación, estos objetos etiquetados se utilizarán como las instancias de entrenamiento de un clasificador que mezcla las características 3D del objeto con la etiqueta asignada por la herramienta externa de etiquetado.



Universitat d'Alacant  
Universidad de Alicante

# Abstract

---

Every day robots are becoming more common in the society. Consequently, they must have certain basic skills in order to interact with humans and the environment. One of these skills is the capacity to understand the places where they are able to move. Computer vision is one of the ways commonly used for achieving this purpose. Current technologies in this field offer outstanding solutions applied to improve data quality every day, therefore producing more accurate results in the analysis of an environment. With this in mind, the main goal of this research is to develop and validate an efficient object-based scene understanding method that will be able to help solve problems related to scene identification for mobile robotics.

We seek to analyze state-of-the-art methods for finding the most suitable one for our goals, as well as to select the kind of data most convenient for dealing with this issue.

Another primary goal of the research is to determine the most suitable data input for analyzing scenes in order to find an accurate representation for the scenes by meaning of semantic labels or point cloud features descriptors.

As a secondary goal we will show the benefits of using semantic descriptors generated with pre-trained models for mapping and scene classification problems, as well as the use of deep learning models in conjunction with 3D features description procedures to build a 3D object classification model that is directly related with the representation goal of this work.

The research described in this thesis was motivated by the need for a robust system capable of understanding the locations where a robot usually

interacts. In the same way, the advent of better computational resources has allowed to implement some already defined techniques that demand high computational capacity and that offer a possible solution for dealing with scene understanding issues. One of these techniques are Convolutional Neural Networks (CNNs). These networks have the capacity of classifying an image based on their visual appearance. Then, they generate a list of lexical labels and the probability for each label, representing the likelihood of the present of an object in the scene. Labels are derived from the training sets that the networks learned to recognize. Therefore, we could use this list of labels and probabilities as an efficient representation of the environment and then assign a semantic category to the regions where a mobile robot is able to navigate, and at the same time construct a semantic or topological map based on this semantic representation of the place.

After analyzing the state-of-the-art in Scene Understanding, we identified a set of approaches in order to develop a robust scene understanding procedure. Among these approaches we identified an almost unexplored gap in the topic of understanding scenes based on objects present in them. Consequently, we propose to perform an experimental study in this approach aimed at finding a way of fully describing a scene considering the objects lying in place.

As the Scene Understanding task involves object detection and annotation, one of the first steps is to determine the kind of data to use as input data in our proposal. With this in mind, our proposal considers to evaluate the use of 3D data. This kind of data suffers from the presence of noise, therefore, we propose to use the Growing Neural Gas (GNG) algorithm to reduce noise effect in the object recognition procedure. GNGs have the capacity to grow and adapt their topology to represent 2D information, producing a smaller representation with a slight noise influence from the input data. Applied to 3D data, the GNG presents a good approach able to tackle with noise.

However, using 3D data poses a set of problems such as the lack of a 3D object dataset with enough models to generalize methods and adapt them to real situations, as well as the fact that processing three-dimensional data is computationally expensive and requires a huge storage space. These

problems led us to explore new approaches for developing object recognition tasks. Therefore, considering the outstanding results obtained by the CNNs in the latest ImageNet challenge, we propose to carry out an evaluation of the former as an object detection system. These networks were initially proposed in the 90s and are nowadays easily implementable due to hardware improvements in the recent years. CNNs have shown satisfying results when they tested in problems such as: detection of objects, pedestrians, traffic signals, sound waves classification, and for medical image processing, among others.

Moreover, an aggregate value of CNNs is the semantic description capabilities produced by the categories/labels that the network is able to identify and that could be translated as a semantic explanation of the input image. Consequently, we propose using the evaluation of these semantic labels as a scene descriptor for building a supervised scene classification model. Having said that, we also propose using semantic descriptors to generate topological maps and test the description capabilities of lexical labels.

In addition, semantic descriptors could be suitable for unsupervised places or environment labeling, so we propose using them to deal with this kind of problem in order to achieve a robust scene labeling method.

Finally, for tackling the object recognition problem we propose to develop an experimental study for unsupervised object labeling. This will be applied to the objects present in a point cloud and labeled using a lexical labeling tool. Then, objects will be used as the training instances of a classifier mixing their 3D features with label assigned by the external tool.



# Contents

---

<b>List of Figures</b>	<b>xxvii</b>
<b>List of Tables</b>	<b>xxxix</b>
<b>List of Algorithms</b>	<b>xxxiii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Introduction . . . . .	1
1.2 Motivation . . . . .	2
1.3 Related works . . . . .	4
1.4 Datasets . . . . .	6
1.4.1 SHOT Dataset . . . . .	6
1.4.2 KTH:IDOL Dataset . . . . .	7
1.4.3 ViDRILO Dataset . . . . .	9
1.5 Deep Learning (DL) . . . . .	11
1.5.1 Convolutional Neural Networks (CNNs) . . . . .	13
1.5.1.1 Convolutional Layers . . . . .	14
1.5.1.2 Activation Functions . . . . .	15
1.5.1.3 Pooling Layers . . . . .	16
1.5.1.4 Fully-Connected Layers . . . . .	17
1.5.1.5 Softmax Layers . . . . .	18
1.5.1.6 Backpropagation . . . . .	18
1.5.1.7 Dropout . . . . .	19
1.5.2 CNN Architectures . . . . .	20
1.6 Deep Learning Frameworks . . . . .	20

1.6.1	Clarifai . . . . .	21
1.6.2	Caffe . . . . .	22
1.6.2.1	Pre-Trained Caffe Models . . . . .	22
1.6.3	CNN architectures for models . . . . .	23
1.6.3.1	AlexNet . . . . .	23
1.6.3.2	GoogLeNet . . . . .	23
1.6.4	Datasets for models . . . . .	24
1.6.4.1	ImageNet 2012 Dataset . . . . .	24
1.6.4.2	Places 205 Dataset . . . . .	25
1.6.4.3	Hybrid MIT Dataset . . . . .	26
1.7	Proposal . . . . .	27
1.8	Goals . . . . .	28
1.9	Structure of the dissertation . . . . .	29
<b>2</b>	<b>Object Recognition in Noisy RGB-D Data</b>	<b>31</b>
2.1	Introduction . . . . .	31
2.2	3D Object Recognition . . . . .	32
2.3	3D Filtering methods . . . . .	33
2.3.1	GNG . . . . .	33
2.3.2	Voxel Grid . . . . .	35
2.4	3D object recognition pipeline . . . . .	38
2.4.1	Normal extraction . . . . .	39
2.4.2	Keypoint detection . . . . .	40
2.4.2.1	Uniform Sampling . . . . .	40
2.4.2.2	Harris 3D . . . . .	40
2.4.2.3	ISS . . . . .	40
2.4.3	Feature description . . . . .	41
2.4.3.1	SHOT . . . . .	41
2.4.3.2	Spin Image . . . . .	41
2.4.3.3	FPFH . . . . .	42
2.4.4	Feature matching . . . . .	42
2.4.5	Clustering features using Geometric Consistency . . . . .	42
2.4.6	ICP Refinement . . . . .	43
2.4.7	Hypothesis Verification . . . . .	43

2.5	Experimentation . . . . .	43
2.5.1	Dataset . . . . .	43
2.5.2	Experimentation setup . . . . .	44
2.6	Results . . . . .	45
2.6.1	Results for SHOT descriptor . . . . .	46
2.6.2	Results for the FPFH feature descriptor . . . . .	50
2.6.3	Results for the Spin Image feature descriptor . . . . .	51
2.6.4	Discussion . . . . .	52
2.7	Conclusions . . . . .	57
<b>3</b>	<b>Scene Classification based on Semantic Labeling</b>	<b>59</b>
3.1	Introduction . . . . .	59
3.2	Scene Classification . . . . .	61
3.3	Scene classification using semantic labels . . . . .	62
3.4	Descriptor generation and description . . . . .	63
3.4.1	Descriptor generation from visual and depth features	64
3.4.1.1	PHOG . . . . .	64
3.4.1.2	GIST . . . . .	65
3.4.1.3	ESF . . . . .	65
3.4.2	Descriptor generation from the Clarifai system . . . . .	67
3.5	Experimental framework . . . . .	68
3.5.1	Evaluation of Clarifai as visual descriptor . . . . .	70
3.5.2	Coping with domain adaptation . . . . .	71
3.5.3	Label Subset Selection . . . . .	74
3.6	Discussion . . . . .	75
3.7	Conclusions and future work . . . . .	78
<b>4</b>	<b>LexToMap: Lexical-based Topological Mapping</b>	<b>81</b>
4.1	Introduction . . . . .	81
4.2	Topological Mapping . . . . .	84
4.3	Lexical-based Image Descriptors . . . . .	85
4.3.1	Image annotation using CNN . . . . .	86
4.3.2	Image/Node Descriptor Similarity Computation . . . . .	88
4.4	LexToMap . . . . .	89
4.5	Experimental Results . . . . .	91



4.5.1	Dataset . . . . .	92
4.5.2	Model Selection . . . . .	92
4.5.3	Topological Map Generation . . . . .	94
4.5.4	Description Capabilities . . . . .	98
4.6	Conclusions and future work . . . . .	99
<b>5</b>	<b>AuSeMap: Automatic Semantic Map Generation</b>	<b>103</b>
5.1	Introduction . . . . .	103
5.2	Semantic Mapping . . . . .	105
5.2.1	Place Recognition for Loop Closing . . . . .	108
5.3	Lexical Labeling using CNNs . . . . .	110
5.4	Bottom-up Aggregation and Similarity Computation . . . . .	110
5.5	Experimentation and Results . . . . .	112
5.5.1	ViDRILO dataset . . . . .	112
5.5.2	Pre-trained models . . . . .	114
5.5.2.1	Baseline results . . . . .	115
5.5.3	Automatic semantic map generation . . . . .	117
5.5.4	Analysis of maps generated . . . . .	119
5.6	Conclusions and Future Work . . . . .	121
<b>6</b>	<b>OReSLab: 3D Object Recognition through CNN Semi-Supervised Labeling</b>	<b>125</b>
6.1	Introduction . . . . .	125
6.2	Related Work . . . . .	128
6.3	OReSLab . . . . .	132
6.3.1	Acquisition Phase . . . . .	134
6.3.2	2D Labeling Phase . . . . .	134
6.3.3	3D Description Phase . . . . .	134
6.3.4	Training and Validation Phase . . . . .	134
6.4	Experimentation . . . . .	135
6.4.1	Object Selection for experimentation . . . . .	135
6.4.2	Experimental Setup . . . . .	137
6.4.3	Baseline Results: proposal validation . . . . .	139
6.4.4	Experimental Results . . . . .	141
6.5	Discussion . . . . .	144

6.6	Conclusions and future work . . . . .	146
<b>7</b>	<b>Conclusions</b>	<b>147</b>
7.1	Conclusions . . . . .	147
7.2	Contributions of the Thesis . . . . .	149
7.3	Publications . . . . .	149
7.4	Future Work . . . . .	151
	<b>Appendices</b>	<b>153</b>
<b>A</b>	<b>Resumen</b>	<b>155</b>
A.1	Introducción . . . . .	155
A.2	Motivación . . . . .	156
A.3	Trabajos Relacionados . . . . .	158
A.4	Conjuntos de Datos . . . . .	161
A.4.1	SHOT . . . . .	161
A.4.2	<i>Dataset</i> KTH:IDOL . . . . .	161
A.4.3	ViDRILO . . . . .	164
A.5	<i>Deep Learning</i> (DL) . . . . .	166
A.5.1	Redes Neuronales Convolucionales . . . . .	168
A.5.1.1	Capas Convolucionales . . . . .	169
A.5.1.2	Funciones de Activación de las Neuronas . . . . .	170
A.5.1.3	Capas de <i>Pooling</i> . . . . .	171
A.5.1.4	Capas Totalmente Conectadas . . . . .	172
A.5.1.5	Capa <i>Softmax</i> . . . . .	174
A.5.1.6	Retro-propagación . . . . .	174
A.5.1.7	<i>Dropout</i> . . . . .	175
A.5.2	Arquitecturas de CNN . . . . .	175
A.6	Sistemas para <i>Deep Learning</i> . . . . .	177
A.6.1	Clarifai . . . . .	177
A.6.2	Caffe . . . . .	178
A.6.2.1	Modelos Pre-Entrenados con Caffe . . . . .	178
A.6.3	Arquitecturas CNN para los modelos . . . . .	179
A.6.3.1	AlexNet . . . . .	179
A.6.3.2	GoogLeNet . . . . .	179

A.6.4	Conjuntos de datos para los modelos . . . . .	180
A.6.4.1	ImageNet 2012 . . . . .	180
A.6.4.2	<i>Places</i> 205 . . . . .	181
A.6.4.3	<i>Hybrid</i> MIT . . . . .	183
A.7	Propuesta . . . . .	183
A.8	Objetivos . . . . .	185
A.9	Conclusiones . . . . .	185
A.10	Contribuciones de la Tesis . . . . .	187
A.11	Publicaciones . . . . .	188
A.12	Trabajo Futuro . . . . .	189

<b>Bibliography</b>	<b>191</b>
---------------------	------------

<b>List of Acronyms</b>	<b>215</b>
-------------------------	------------



Universitat d'Alacant  
Universidad de Alicante

# List of Figures

---

1.1	Objects from the SHOT dataset. . . . .	7
1.2	Scenes from the SHOT dataset. . . . .	8
1.3	KTH-IDOL 2 information. . . . .	8
1.4	Images from the KTH-IDOL 2 dataset. . . . .	9
1.5	Changes produced by the human interaction. . . . .	10
1.6	Frame of the ViDRILO dataset. . . . .	11
1.7	Categories in ViDRILO dataset . . . . .	12
1.8	Architecture of a standard CNN. . . . .	13
1.9	Convolution example. . . . .	15
1.10	Standard activation functions . . . . .	16
1.11	Max Pooling operation. . . . .	16
1.12	Multilayer perceptron module. . . . .	17
1.13	Fully Connected Layers. . . . .	18
1.14	CNN Layer Visualization . . . . .	21
1.15	Inception Module. . . . .	24
2.1	GNG scheme. . . . .	37
2.2	Object represented using the GNG. . . . .	38
2.3	Object represented using Voxel Grid. . . . .	38
2.4	Recognition pipeline scheme. . . . .	39
2.5	Recognition result obtained by the pipeline. . . . .	45
2.6	Mean results charts for SHOT. . . . .	49
2.7	Mean results charts for FPFH. . . . .	53
2.8	Mean results charts for Spin Image . . . . .	56
2.9	Results for the experiments by detector, descriptor and filter. . . . .	56

3.1	Clarifai results for a ViDRILO image. . . . .	61
3.2	Methodology in overall pipeline. . . . .	64
3.3	Generation of the PHOG Descriptor. . . . .	65
3.4	Generation of the GIST Descriptor. . . . .	66
3.5	Generation of the ESF Descriptor. . . . .	66
3.6	Cloud tag for the labels in the codebook . . . . .	69
3.7	Accuracy obtained for the experiments. . . . .	71
3.8	Average accuracy over training/test combinations. . . . .	72
3.9	Accuracy obtained for experiments that use Sequence 5. . . . .	75
3.10	Average ranking for all classifier and descriptor combinations. . . . .	76
3.11	Accuracy of the Clarifai Descriptors using a Subset Variable Selection Process . . . . .	77
3.12	Frequency obtained by the labels in ViDRILO. . . . .	77
3.13	Images for the tags in the Clarifai descriptor. . . . .	78
4.1	Map types examples. . . . .	82
4.2	Aggregation process example. . . . .	88
4.3	Scheme of the LexToMap proposal. . . . .	91
4.4	Intra-cluster spatial evolution for CNN models. . . . .	93
4.5	Ranking comparison of 7 CNN models. . . . .	95
4.6	Topological maps generated in the experiments. . . . .	96
4.7	Identified transition in the mapping. . . . .	98
4.8	Not identified transition in the mapping. . . . .	98
4.9	Location description using lexical labels. . . . .	100
4.10	Locations representative of some lexical labels. . . . .	100
4.11	Object appearance in topological maps described with color codes. . . . .	101
5.1	Overall scheme of the proposal. . . . .	105
5.2	Ground truth in the ViDRILO Dataset. . . . .	114
5.3	Comparative of images belonging to different locations with same category. . . . .	115
5.4	Distance Matrix for the categories in ViDRILO dataset. . . . .	116
5.5	Quantitative metric generation. . . . .	119

5.6	Semantic map generated from Sequence 1 with the Places-GoogLeNet model. . . . .	122
5.7	Semantic map generated from Sequence 2 with the Places-GoogLeNet model. . . . .	123
6.1	Proposal flowchart. . . . .	127
6.2	Example of the Acquisition (left) and 2D Labeling Phase (right) of the proposal. . . . .	135
6.3	Cloud tags representing the labeling tool high (left) and low (right) assertion rate for object instances selected for the study. . . . .	136
6.4	Images used for the experiment . . . . .	137
6.5	Top 1 labeling generated by the labeling tool. . . . .	139
6.6	Confusion Matrix (left) and set distribution (right) for the baseline set experiment. . . . .	140
6.7	Accuracy obtained by every combination. . . . .	142
A.1	Objetos presentes en SHOT. . . . .	162
A.2	Escenas en SHOT. . . . .	163
A.3	Información KTH-IDOL 2. . . . .	163
A.4	Imágenes del conjunto de datos KTH-IDOL 2. . . . .	164
A.5	Cambios producidos por la interacción humana. . . . .	165
A.6	Imagen de ViDRILO. . . . .	166
A.7	Categorías en ViDRILO . . . . .	167
A.8	Arquitectura estándar de una CNN. . . . .	168
A.9	Ejemplo de convolución. . . . .	170
A.10	Funciones de Activación estándar . . . . .	171
A.11	<i>Max Pooling</i> . . . . .	172
A.12	Perceptrón Multicapa. . . . .	173
A.13	Capas totalmente conectadas. . . . .	174
A.14	Visualización por capa de una CNN . . . . .	176
A.15	Módulo <i>Inception</i> . . . . .	180



# List of Tables

---

1.1	Overall ViDRILO sequence distribution. . . . .	11
1.2	Pre-trained CNN models . . . . .	23
1.3	Images from the ImageNet Dataset . . . . .	25
1.4	Images from the Places 205 Dataset . . . . .	26
2.1	Experiment combinations. . . . .	44
2.2	Results for SHOT and Uniform Sampling. . . . .	46
2.3	Results for SHOT and Harris 3D. . . . .	47
2.4	Results for SHOT and ISS. . . . .	48
2.5	Mean results for SHOT. . . . .	48
2.6	Average results grouped by the cloud size for SHOT. . . . .	49
2.7	Results for FPFH and Uniform Sampling. . . . .	50
2.8	Results for FPFH and Harris 3D. . . . .	51
2.9	Results for FPFH and ISS. . . . .	51
2.10	Mean results for FPFH. . . . .	52
2.11	Average results grouped by the cloud size for FPFH. . . . .	52
2.12	Results for Spin Image and Uniform Sampling. . . . .	53
2.13	Results Spin Image for and Harris 3D. . . . .	54
2.14	Results for Spin Image and ISS. . . . .	54
2.15	Mean results for Spin Image. . . . .	55
2.16	Average results grouped by the cloud size for Spin Image. . . . .	55
2.17	Grouped results of experimentation. . . . .	55
3.1	Clarifai descriptor generation. . . . .	68



3.2	Post-hoc comparison for the descriptor/classifier combinations. . . . .	73
3.3	Comparison of images belonging to categories in ViDRILO. . . . .	74
4.1	Intra-cluster variance for 7 CNN models. . . . .	94
5.1	Classification of ViDRILO images and distribution for sequences 1 and 2. . . . .	113
5.2	Number of clusters generated. . . . .	117
5.3	Distances in the cluster distributions. . . . .	120
6.1	Labeling Comparison . . . . .	133
6.2	Experimental Setup . . . . .	138
6.3	Success case from the baseline experiment . . . . .	141
6.4	Fail case from the baseline experiment . . . . .	141
6.5	Success case from the lowest accuracy experiment . . . . .	143
6.6	Fail case from the lowest accuracy experiment . . . . .	143
6.7	Training Instances Comparison . . . . .	144
6.8	Common instances in the training set of the lowest accuracy experiments . . . . .	145
A.1	Distribución de las Secuencias en ViDRILO. . . . .	166
A.2	Modelos CNN pre-entrenados . . . . .	179
A.3	Imágenes de ImageNet . . . . .	181
A.4	Imágenes de <i>Places</i> 205 . . . . .	182

# List of Algorithms

---

2.1	Pseudo-code of the GNG algorithm. . . . .	36
4.1	LexToMap: Lexical-based Topological Mapping . . . . .	90
5.1	Hierarchical clustering for bottom-up aggregation . . . . .	112



Universitat d'Alacant  
Universidad de Alicante



# Introduction

---

This first chapter introduces the main topic of this thesis. The chapter is organized as follows: Section 1.1 establishes the framework for the research activity proposed in this thesis. Section 1.2 introduces the motivation of this work. Section 1.3 is a state-of-the-art analysis of the existing classification and mapping procedures. Section 1.4 shows the datasets that this thesis will use in the different experiments. Section 1.5 defines deep learning as an approach to tackle classification problems, which are presented in this thesis. Section 1.6 describes the deep learning frameworks used in this work. In Section 1.7 the proposal developed in this thesis is introduced, and Section 1.8 presents the main goals of this work. Finally, Section 1.9 provides a detailed analysis of the dissertation's structure.

## 1.1 Introduction

In this doctoral thesis, theoretical and practical research was conducted to explore the issue of finding an accurate representation model to improve image understanding. We aim to focus on the image representation methods that better describe the content of scenes in order to semantically classify sets of images. We study novel methods for describing scenes based on semantic labels, using deep learning techniques applied to data captured from the real world. Moreover, the selected representation method will be tested for semantic and topological mapping problems, two areas where

traditional methods are still encountering problems that need to be solved.

This thesis has been conducted within the framework of the following projects:

- *RETOGAR: Retorno al hogar. Sistema de mejora de la autonomía de personas con daño cerebral adquirido y dependientes en su integración en la sociedad.* Founded by the Spanish Ministry of Economy and supported FEDER funds. DPI2016-76515-R.
- *SIRMAVED: Development of a comprehensive robotic system for monitoring and interaction for people with acquired brain damage and dependent people.* . Founded by the Spanish Ministry of Economy and supported by FEDER funds. DPI2013-40534-R.

and under the following grant:

- IFARHU grant 8-2014-166, of the Doctoral Program IFARHU-UTP 2014 of the Republic of Panamá.

Moreover, part of the work presented in this thesis was done during my stay in the Data Intelligence Group at University Jean Monnet in Saint-Etienne, France. This three month stay was funded by the University of Alicante PhD School. The work carried out during my stay was partially led by professor Marc Sebban.

## 1.2 Motivation

This document is the result of the doctoral studies carried out within the framework of the PhD in Computer Engineering, completed between 2014 and 2017 at the University of Alicante University Institute for Computing Research (Spain). This thesis was derived from a four-year PhD fellowship awarded to me by the Government of the Republic of Panamá in collaboration with the *Universidad Tecnológica de Panamá* and the *Instituto para la Formación y Aprovechamiento de los Recursos Humanos* (IFARHU).

The motivation for this thesis project arises from the participation and collaboration in many different projects related with computer vision and mobile robotics issues.

The Simultaneous Localization and Mapping (SLAM) method is a commonly used technique in the mobile robotics field which has been extensively studied since it was originally developed by [Leonard and Durrant-Whyte, 1991] based on the earlier work by [Smith et al., 1987]. This technique focuses on the locating the robot in a environment as well as mapping said environment, in order to allow the robot to navigate it using the created map.

However, SLAMs methods could not tell a robot the kind of location where it is located. This information must be acquired by analyzing the environment, and producing a semantic description of the place. Traditionally the semantic description of places has been carried out by identifying objects in the scene. Hence why nowadays the semantic description is usually related to object recognition as well as to scene understanding. However, object recognition still has to deal with several problems such as occlusions, rotations and partial views of the objects, among others. Therefore, this research line demands new strategies to deal with the information. Semantic categorization of places is necessary for interactive robots. These robots should be able to understand human orders related to rooms in an environment, i.e. a kitchen in a house. Thus, robots require a method to infer their location as well as infer a new location at the time that the robot is watching a new one. In this thesis we focus on finding an accurate environment representation in order to achieve a semantic description of places.

In recent years the advent of better computational resources such as GP-GPU graphic cards, better CPUs, and increased memory, has made it possible to implement some already-defined techniques that demand high computational capacity. Among them we could mention Convolutional Neural Networks (CNNs). NNs are able to classify an image based on their visual appearance. On the other hand, CNNs allow us to get a list of probabilities that represent the likelihood of the presence of an object in the scene. Possible objects on the list are derived from the training sets that the networks learned to recognize. Therefore, this list of probabilities allows to describe an environment. In this thesis we focus on using this list of probabilities as an efficient representation of the environment and

then assigning a semantic category to the regions where a mobile robot is able to navigate using a map that could have been constructed using an SLAM approach. In this thesis we also constructed topological and semantic maps using the semantic descriptors generated by the CNNs.

During my collaboration with the Data Intelligence Group I worked on the design of a semantic representation of the objects that are present in a scene, based on deep learning techniques. This representation was required to develop a real-time system able to detect the kind of place where the robot was located.

### 1.3 Related works

In this thesis we are dealing with problems related to scene understanding guided by object recognition.

Commonly, scene understanding has been addressed by pointing out objects that are present in the scenes. As proposed in [Li et al., 2009] authors developed a framework that uses tagged images from Flickr.com and is able to classify, annotate and segment images of a variety of scenes. The framework uses a hierarchical model to unify information from several levels (object, patch and scene).

The work presented in [Liao et al., 2016] exploits CNNs' capabilities for learning objects features in order to construct a CNN scene classification model with regularization of segmentation (SS-CNN). Here, authors take advantage of the interaction of objects and scenes to perform a semantic segmentation based on prior information about objects occurrence. The object-level information is learned in early stages of the process because said stages are related to the features learned for scene classification.

The work put forward by [Nguyen et al., 2016] proposes to use a Multi-Level CNN to extract features of an image. Features are extracted by taking the output of the last hidden layer in the network, in other words, the fully-connected layer just before the final classification layer. CNN features are extracted in two levels: global and regional. Firstly, global features are used to seek for similar images in the retrieval database. Secondly, once the similar images have been found, regional features are computed from

super pixels. Regional features also include a set of hand-crafted features (shape, texture, etc). Then, using the combination of regional features, it computes similarities to calculate a probability value for a set of labels, with the aim of producing image parsing. In a similar way, the CollageParsing [Tung and Little, 2016] algorithm uses Markov Random Fields (MRF) to find related images, but instead of using super pixel comparison to do a label transfer, authors define content-adaptive windows. These windows are used to compute unary potentials, by matching similar contents in the retrieval set of the image.

In relation to benchmarks suites for scene understanding, SUN RGB-D [Song et al., 2015], proposes a benchmark designed for testing scene understanding approaches. Here, images contain annotations, bounding boxes and polygons. This has been evaluated in several tasks such as scene categorization, semantic segmentation, object detection, object orientation, room layout estimation, and total scene understanding. On the other hand, the ScanNet [Handa et al., 2016] framework consists of annotated synthetics scenes that includes objects and labels for the scenes. These scenes contain objects sampled from several datasets like ModelNet, as well as textures from OpenSurfaces and ArchiveTextures. The framework allows users to generate annotated scenes based on the objects available in the dataset. The SUN Attribute Database [Patterson et al., 2014] consists of a set of images that have been annotated (by humans) with attributes, belonging to 707 different categories. This work studies the give-and-take between categories and attributes and tests the dataset in tasks such as scene classification, zero shot learning, and semantic image search, among others. The Places Dataset [Zhou et al., 2014] is a scene-centric dataset made up of a huge number of images and annotated with their categories that has been tested both in scene classification tasks and in the use of deep learning to build classification models.

Scene understanding has been dealt with from several approaches, such as the Hierarchical Bayesian Models in [Steinberg et al., 2015], which focus on using unsupervised or visual-data only techniques that could be applied in order to get appropriate scene understanding. The work proposed by [Li et al., 2015] plans to use location-sensitive hashing (LSH) to overcome



visual polysemia and concept polymorphism (VPCP) and improve scene understanding in large images. The proposal by [Redondo-Cabrera et al., 2014] introduces a 3D Bag-Of-Words model for categorization of objects and scenes. The model uses quantized local 3D descriptors obtained from point clouds.

The proposal by [Vaca-Castano et al., 2017] studies how the objects are related to a scene in egocentric vision constraints. The work proposes improving object detection by predicting a scene category first, and then computing the objects that could be present in the image according to the category. Authors also present a framework with which using Long Short-Term Memory networks in the preliminary scene category is not necessary.

[Tung and Little, 2015] employ a concatenated version of object detectors produced by Never-Ending Image Learning (NEIL) and ImageNet models to learn the objects and their relationship in the scenes. This concatenation was developed to improve attribute recognition in images.

## 1.4 Datasets

This Section introduces the reader to the datasets which will be used as validation sets for our proposed methods. Several public datasets have been used to validate the proposed methods.

### 1.4.1 SHOT Dataset

Aimed at testing object recognition approaches as a step in scene understanding, we experimented with 3D object recognition methods. We used the SHOT Dataset<sup>1</sup> (University of Bologna)[Tombari et al., 2010][Tombari and Salti, 2011] [Computer Vision LAB, 2013]. This has been acquired by means of the Spacetime Stereo (STS) technique, and it consists of 7 models with different views each, and 17 scenes for a total of 49 object recognition instances. Figure 1.1 shows some objects of the dataset whereas Figure 1.2 exposes some scenes where objects appear. Each scene contains a random number of objects, and the objects could appear in several positions and views.

---

<sup>1</sup><http://www.vision.deis.unibo.it/research/80-shot>



Figure 1.1: Examples of objects in the SHOT dataset.

### 1.4.2 KTH:IDOL Dataset

We chose the KTH-IDOL 2 dataset [Luo et al., 2007] for the evaluation of our topological mapping proposal. The Image Database for rObot Localization (IDOL) is an indoor dataset that provides sequences of perspective images taken under three different lighting conditions: sunny, cloudy and night. These sequences were generated using two different robot platforms, namely Minnie (PeopleBot) and Dumbo (PowerBot), which were controlled by a human operator. The ground truth in the dataset includes the following information for each image: the semantic category of the room where the image was taken, the timestamp, and the pose of the robot ( $\langle x, y, \theta \rangle$ ) when the shot was taken. There are five different room categories: corridor (CR), kitchen (KT), one-person office (1-PO), two-persons office (2-PO), and printer area (PA). The dataset includes four different sequences for each combination of robot and lighting conditions. From all these sequences, we selected the twelve with Minnie, where camera position (around one meter above the floor) is more similar to that of



Figure 1.2: Examples of the scenes from the SHOT dataset.

most current mobile robot platforms.

Lighting Cond.	Cloudy	Night	Sunny
Sequences	1,2,3,4	1,2,3,4	1,2,3,4
#Images CR	1,632	1,704	1,582
#Images 1-PO	458	522	468
#Images 2-PO	518	633	518
#Images KT	656	611	556
#Images PA	488	534	482
#Images	3,752	4,004	3,606

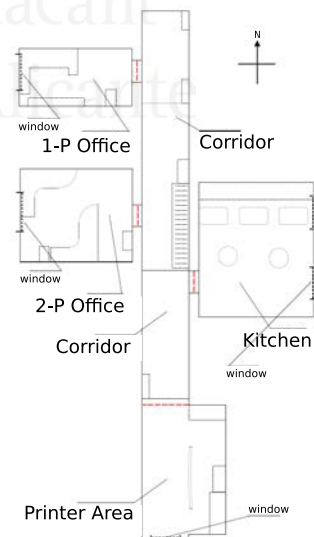


Figure 1.3: KTH-IDOL 2 information: Image distribution (left) and environment (right).

The number of images in the dataset for each set of lighting conditions and semantic category, as well as the map used for the shoot, are shown in Figure 1.3. Image distribution is clearly unbalanced as most of the images belong to the Corridor category. Sequences 3-4 were taken six months later than sequences 1-2, which introduced small environmental variations due to human activity. Figure 1.4 presents 15 exemplar images from the dataset.

These examples show how visual representations are affected by lighting conditions. Moreover, Figure 1.5 illustrates the effect of human activity over the same locations in the environment.

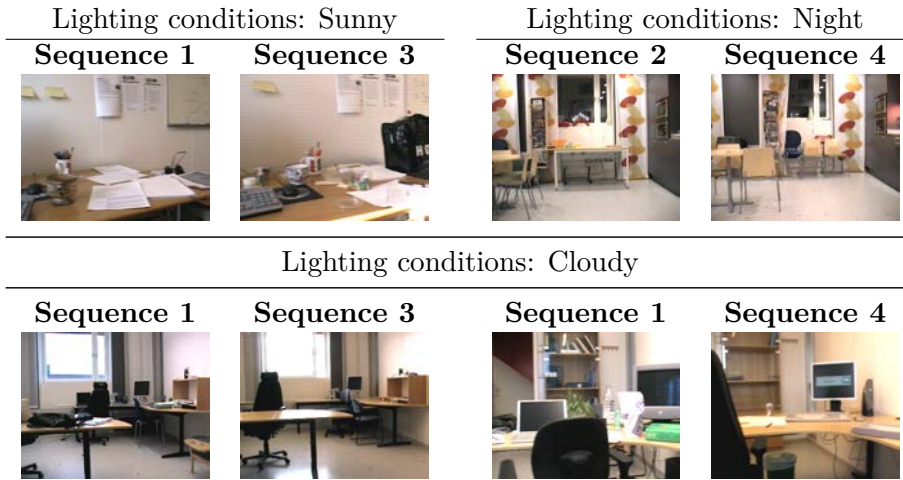


**Figure 1.4:** Exemplars from the KTH-IDOL 2 dataset taken under three lighting conditions (rows) within five different room categories (columns).

### 1.4.3 ViDRILO Dataset

The Visual and Depth Robot Indoor Localization with Objects information Dataset (also known as ViDRILO) [Martínez-Gomez et al., 2015] has been created for the Robot Vision ImageClef<sup>2</sup> Challenge. This challenge addresses the problem of semantic place classification using visual and depth information including object recognition. The ViDRILO dataset has been chosen as input data for scene classification and semantic mapping proposals.

<sup>2</sup><http://imageclef.org>



**Figure 1.5:** Images illustrating changes in the environment caused by human activity.

The main characteristics of this dataset<sup>3</sup>, which provides five different sequences of RGB-D images captured by a mobile robot within an indoor office environment, are shown in Table 1.1. The dataset was acquired over a span of twelve months in order to incorporate variations due to human activity over time.

The whole dataset was acquired in two office buildings (Polytechnic School 2 and 3) at the University of Alicante (Spain), using a Powerbot robot with a Kinect on-board. The robot has a sweeping unit equipped with a 2D Sick laser. A Kinect camera was placed on top of the sweeping unit, thus providing a total height of 90cm. The robot was tele-operated with a joystick during the complete path, at an average speed of 0.3m/s. The dataset contains both the RGB image (color image) and the 3D colored point cloud of various sequences taken in different rooms with a different distribution of objects. Figure 1.6 shows an example of a secretary room of the provided data.

Each RGB-D image is annotated with the semantic category of the scene in which it was acquired, from a set of ten different room categories. Different sequences from the ViDRILO dataset were used as the benchmark

<sup>3</sup>Available for download at <http://www.rovit.ua.es/dataset/vidriilo/>



**Figure 1.6:** Example of one frame data from the ViDRILO Dataset. Color image on the left and point cloud on the right.

**Table 1.1:** Overall ViDRILO sequence distribution.

Sequence	Number of Frames	Floors imaged	Dark Rooms	Time Span	Building
Sequence 1	2,389	1st,2nd	0/18	0 months	A
Sequence 2	4,579	1st,2nd	0/18	0 months	A
Sequence 3	2,248	2nd	4/13	3 months	A
Sequence 4	4,826	1st,2nd	6/18	6 months	A
Sequence 5	8,412	1st,2nd	0/20	12 months	B

in the RobotVision challenge at the most recent editions of the ImageCLEF competition [Martínez-Gómez et al., 2015]. We opted for this dataset because: (a) it provides sequences of RGB-D images acquired with temporal continuity; (b) the scenes are semantically labeled; and (c) spatially separated areas can share the same semantic category, allowing generalization. In Figure 1.7, representative images for all ten room categories are shown.

## 1.5 Deep Learning (DL)

The emergence of DL in the robotics community has opened up new research opportunities in the last few years. In addition to model generation for solving open problems [Bo et al., 2013, Neverova et al., 2014], the release of pre-trained models allows for a direct application of the generated DL systems [Rangel et al., 2016a]. This is possible thanks to the existence of modular DL frameworks such as Caffe [Jia et al., 2014]. The direct application of pre-trained models avoids the computational requirements for





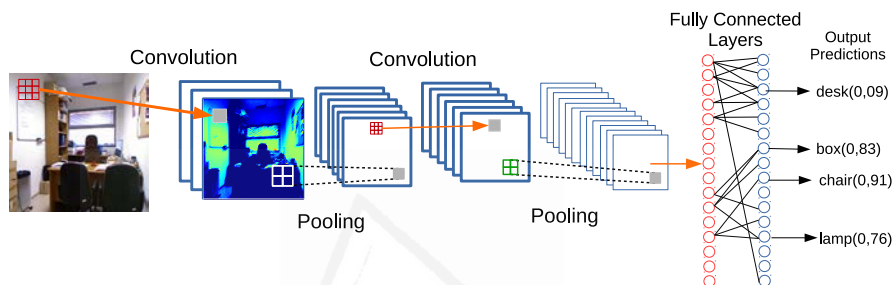
**Figure 1.7:** Exemplar visual images for the 10 room categories in ViDRiLO.

learning them: long learning/training time (even using GPU processing) and massive data storage for training data. Among the existing DL models, we should point out those generated from images categorized with generalist and heterogeneous lexical labels [Krizhevsky et al., 2012, Zhou et al., 2014]. The use of these models lets any computer vision system annotate input images with a set of lexical labels describing their content, as it has been recently shown in [Carneiro et al., 2015, Murthy et al., 2015, Rangel et al., 2016a].

The use of DL is considered to be a remarkable milestone in the research areas of computer vision and robotics [LeCun et al., 2010]. DL provides classifiers that are able to not only classify data, but also automatically extract intermediate features. This technique has been applied to image tagging with surprising results. For instance, the Clarifai team won the 2013 Imagenet competition [Russakovsky et al., 2015] by using Convolutional Neural Networks [Krizhevsky et al., 2012]. In addition to very large amounts of annotated data for training, DL requires high level processing capabilities for learning. While these two requirements are not always met, we can take advantage of some existing solutions that provide DL capabilities through application programming interfaces (APIs), such as Clarifai or Caffe, that will be described later.

### 1.5.1 Convolutional Neural Networks (CNNs)

CNNs are defined as hierarchical machine learning models, which learn complex images representations from large volumes of annotated data [Krizhevsky et al., 2012]. They use multiple layers of basic transformations that finally generate a highly sophisticated representation of the image [Clarifai, 2015]. Figure 1.8 shows the basic structure of a CNN architecture.



**Figure 1.8:** Architecture of a standard CNN.

CNNs are a kind of neural network that focus on processing images, in order to obtain a set of features that fully describe them, through a concatenation of several types of processing layers.

These networks could be separated in two different sections, namely: feature extraction and classification. Firstly, feature extraction computes the features of the images using convolutional and pooling layers. Next, the second section of the network is responsible for computing the output of the net based on the features extracted in the first section. This output is calculated using a set of fully connected layers at the end of the network. The output size is determined by the amount of categories that the network was trained for recognizing.

The use of CNNs in classification problems requires previous steps in order to produce a classifier. The first phase is to gather and order the information (images for CNNs) that will be used for training the model, in other words the training set. Images are grouped in several categories according to the classes which the CNN must recognize. Then, the second phase is the training procedure. This is a crucial step that will define the



accuracy of the classification model. The training phase of a CNN works by analyzing the whole set of gathered images in the training set. Each image will be passed through the layers of the CNN in order to teach the network to match features with a corresponding category.

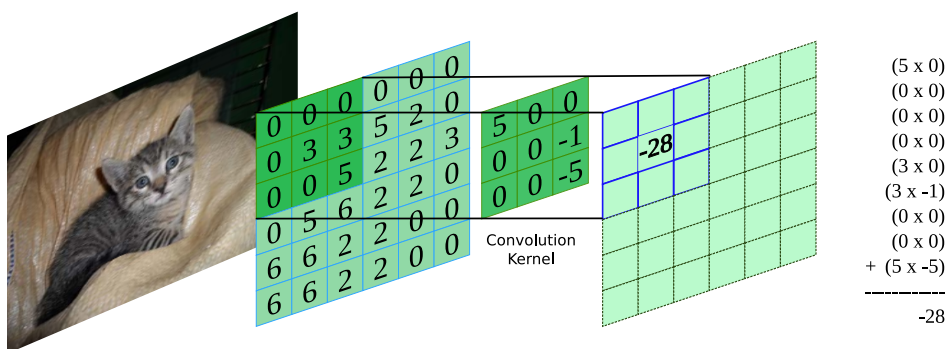
The layers between the first (input layer) and the last (output layer) are known as hidden layers of the neural network. These layers vary in number according to the architecture of the network and the features of the images to be processed.

### 1.5.1.1 Convolutional Layers

Usually, the first layer of a CNN is a convolutional layer. This kind of layer works by applying a convolutional filter (kernel) to the input. This filter is applied as a sliding window over the image. Consequently, the image is partially analyzed in chunks of a defined size. Filters, as images, are defined using two values (width and height), usually with the same size. Nowadays, they have three channels for color information, but at the beginning the input was in gray scale images. Then, a filter size could be defined as  $5 \times 5 \times 3$ . Hence, when a filter is applied over a portion of the image, an element-wise multiplication is carried out and their results are summed up to get the kernel response to that image portion. This multiplication (kernel application) is repeated until the entire image has been analyzed. The output of the kernel operation produces an activation map of the input data where this map is composed of the whole set of kernel responses of the image. This map usually has a smaller size than the original image, although other approaches could be applied to manage the image borders. Kernels usually include a stride value that determines how many pixels the kernel will move in the defined direction.

A convolutional layer could have several kinds of filters, then the output of a convolutional layer it is not only one activation map, but rather a set of these, one for every kernel in the convolutional layer. Figure 1.9 shows the convolution operation for an example kernel.

In convolutional layers, kernels seek the appearing of specific features in the input image. These kernels usually begin by identifying simple features, such as points or curves in the first convolutional layers, then



**Figure 1.9:** Convolution example.

subsequent layers could be prepared to identify groups of these features and then, successively, find more complex or high-level features such as boxes or circles.

### 1.5.1.2 Activation Functions

ReLU or Rectified Linear Units use the output of a convolutional layer and then apply a non-saturating activation function  $f(x) = \max(0, x)$  (Figure 1.10 left). The use of ReLUs in the training processes improves training, thus reducing the time required to train models. It also helps solve the vanishing gradient problem of the networks. The objective of this function is to introduce nonlinearity into the system, due to the fact that convolutional layers have been computing linear operations (element wise multiplications and summations). The activation functions help raise the non-linear attributes of the model and the network. This takes place without changing the receptive field of the convolutional layers. In the past, the modernization of a neuron's output was in charge of saturating non-linearity functions, specifically to the hyperbolic tangent  $f(x) = \tanh(x)$  (Figure 1.10 center), and the sigmoid function  $f(x) = (1 + e^{-x})^{-1}$  (Figure 1.10 right). These are slower than ReLU units, in terms of training time when using the gradient descent optimization algorithm.

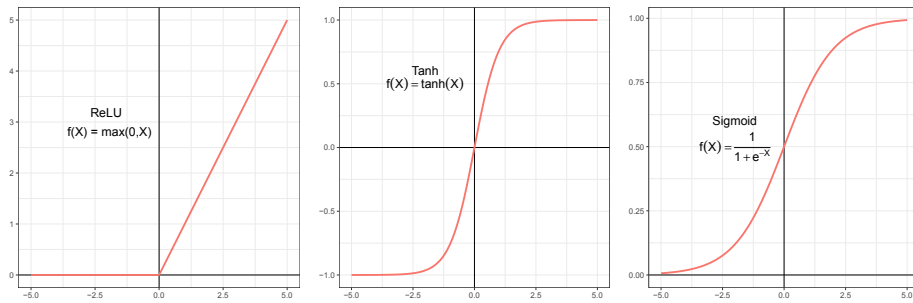


Figure 1.10: Standard activation functions

### 1.5.1.3 Pooling Layers

Pooling acts as a sub-sampling procedure. These layers take each of the activation maps produced by a convolutional layer as input. Much like convolutional layers, this layer works as a sliding windows with a defined size. Then, the layer works by moving the windows over the activation map, selecting a group of pixels from the activation map and then selecting the maximum, minimum or mean value of the group, replacing the group by that value. Then, the output of the pooling layer is a smaller activation map. Figure 1.11 shows a max pooling example operation in one slide of the activation map of a convolutional layer.

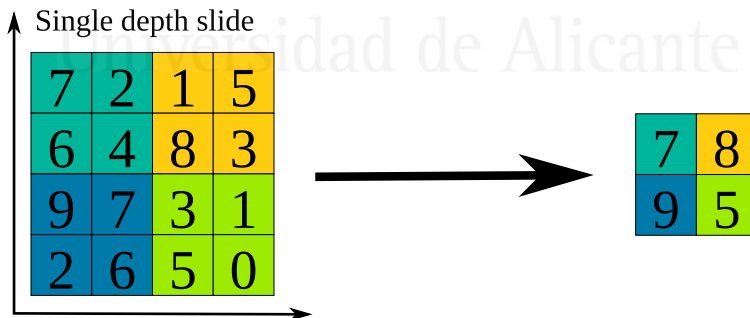


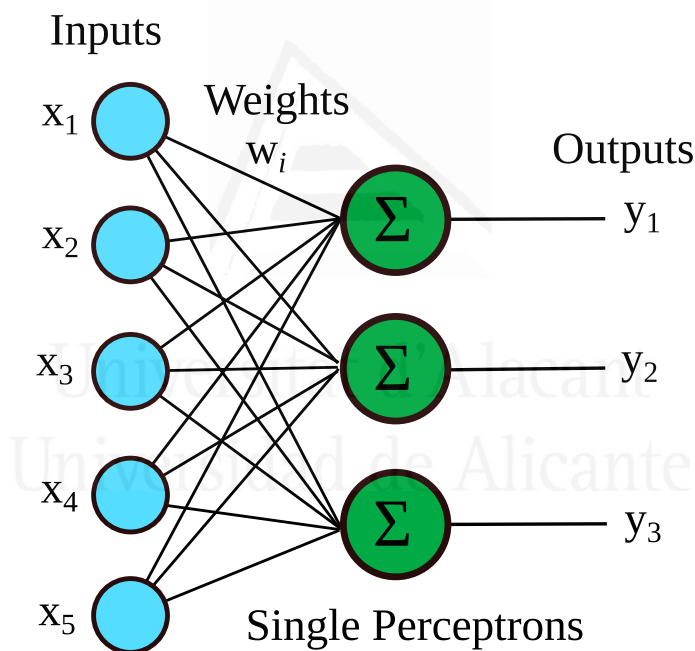
Figure 1.11: Max Pooling operation.

This sequence of convolution and pooling is the base of the CNNs working scheme, but a pooling layer does not always follow a convolutional layer. Some architectures include several convolutional layers, one after another. The CNN architecture will define how the interaction between

layers will occur.

#### 1.5.1.4 Fully-Connected Layers

In a convolutional neural network the last layers correspond to the fully-connected layers that are in charge of generating the output of the network. This kind of layer works similarly to a multilayer perceptron (Figure 1.12), by producing a  $n$ -dimensional vector as output. The vector size will depend on the amount of categories the CNN could have to choose from. For instance, an alphabet classifier will only produce 26 outputs, one per each letter.

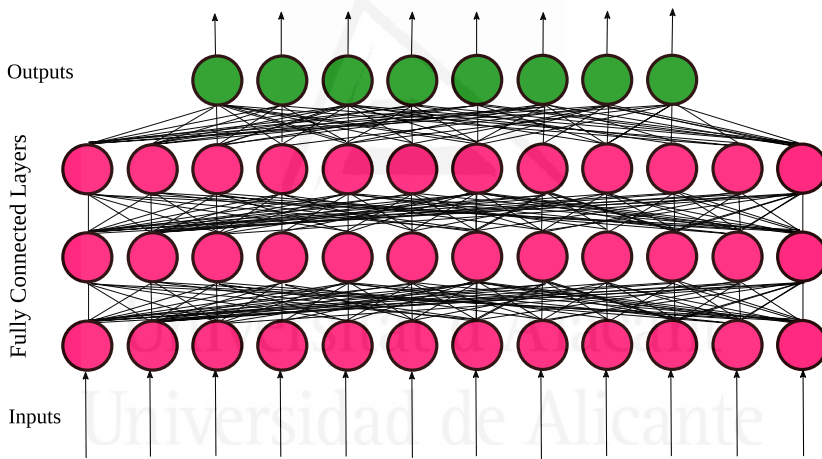


**Figure 1.12:** Multilayer perceptron module.

These layers take the output of the previous layers (convolutional, pooling, or an activation function layer), and then use all those activation maps as their input. Next, each activation map is connected to every neuron of the fully-connected layer and multiplied by a weight in order to produce the output, determining which high-level features most correlate to a particular category.

Basically, fully-connected Layers seek strong correlations among high level features and a particular category with their particular weights. Therefore, once the products between the previous layer and the weights are computed, the categories will obtain the correct probability.

A fully connected layer consists of several multilayer perceptron modules, these modules could be described as a mathematical function that maps some sets of input values (activation maps in CNNs) to output values (categories in CNNs). The function is formed by composing many simpler functions. We can think of each application of a different mathematical function as providing a new representation of the input [Goodfellow et al., 2016]. Figure 1.13 shows a usual fully connected structure where three of the layers are connected to produce the output of the network.



**Figure 1.13:** Fully Connected Layers.

### 1.5.1.5 Softmax Layers

Softmax layer takes the output of fully-connected layers and express these values as a likelihood distribution. Then, the summarized value for the network output will always be equal to 1.0.

### 1.5.1.6 Backpropagation

In CNNs the different hidden layers produce an output that is used by the next layer in the architecture. These outputs are used for the

neuron's layers in order to produce their own output. The propagation of these outputs between layers generates the response of the CNN. The response is produced by a successive multiplications of the outputs by a weight value that represents the apportion of every neuron to the response. However, during the training stage, Backpropagation [Rumelhart et al., 1986] algorithm is in charge of calculating how well the CNN response fits the desired response for the training instance. This algorithm calculates an error value for each output. Then, this value is back propagated to the hidden layers of the CNN, and every hidden layer will be affected by the error value depending on how the neurons contributed to the output using a weight value. Following the same strategy, the weights of the kernels in the convolutional layers are updated, according to the calculated error. The backpropagation is repeated for every layer in the network as well as with every image in the training set. This process ensures that different neurons can specialize on detect different features, and then in the testing stage these neurons will be activated for an unseen image [Goodfellow et al., 2016].

#### 1.5.1.7 Dropout

Dropout [Srivastava et al., 2014] is a regularization method for improving neural networks performance by reducing overfitting. Backpropagation algorithm could lead to the development of adaptations that work for training data, but without the capacity of generalize to unseen data. Dropout proposes to randomly deactivate some output neurons in order to break these adaptations make the presence of any particular hidden unit unreliable. Dropout works by randomly dropping units and their connections from the net during the training of the neural net. The idea behind this is to avoid that units adapt to the training data too much. [Goodfellow et al., 2016] defines dropout as: the function that trains the ensemble consisting of all sub-networks that can be formed by removing non-output units from an underlying base network. This process could be done by taking the output of the neuron and multiplying it by zero.

### 1.5.2 CNN Architectures

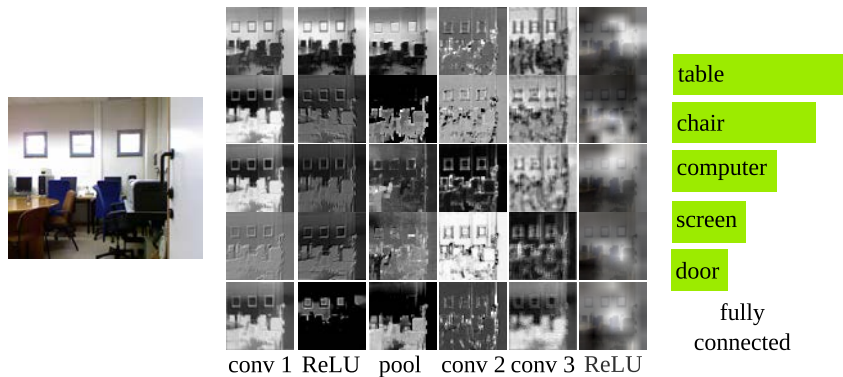
Every CNN has its own architecture and this can be described as a successive application of non-similar filters and layers. Each layer is able to recognize a specific characteristic of an image, from a low detail level (pixel) to more detailed patterns (shapes). Every CNN involves the interaction of several of these layers, and according to their function layers could be convolutional, pooling, or fully connected, among others. Now there are, a vast number of different architectures. Each one has certain distinctive properties, such as the amount of convolution layers. GoogLeNet [Szegedy et al., 2014] and AlexNet [Krizhevsky et al., 2012] are two widely known architectures for image description/classification in the field of CNN.

A CNN model is defined by a combination of architecture and dataset that was used for training the architecture. The last layer of every CNN model is responsible for classification, and it maps the values computed by former layers to a size-limited output. Every architecture must specify the number of outputs to be produced by the model, which corresponds to the dimensionality of the classification problem. Each model encodes the information retrieved from a selected dataset, so the number of outputs is determined by the dataset used in the training process.

Each layer of a CNN produces modifications over the output of the previous layer. Figure 1.14 shows the output produced by some layers of a CNN design, as well as the Visualization of these modifications. In the last ReLU layer it is possible to distinguish how the CNN is able to highlight regions that contain objects in the image.

## 1.6 Deep Learning Frameworks

This section introduces the Deep Learning frameworks used in the experiments carried out during the develop of this thesis.



**Figure 1.14:** Results produced by several layers of a CNN.

### 1.6.1 Clarifai

Clarifai<sup>4</sup> is one of the well-known systems offering remote image tagging. Specifically, any input image is labeled with the semantic categories that best describe the image content. Clarifai technology relies on the use of CNNs [Krizhevsky et al., 2012] to process an image, and then generates a list of tags describing the image. The Clarifai approach was firstly proposed as a solution to the ImageNet [Russakovsky et al., 2015] classification challenge [Krizhevsky et al., 2012] in 2013, where the system produced one of the top-5 results. However, the Clarifai service is now a closed system whose details about the datasets used for training (which determine the dimension of the decision layer) and internal architecture are not provided. Therefore, we state the maximum number of the dimension for the extracted descriptors in a preliminary stage at which we discover all the annotations that are extracted from the dataset. This is similar to the codebook identification when applying a Bag-of-Words approach [Filliat, 2007].

Clarifai works through the analysis of images to produce a list of descriptive tags that are representative of a given image. For each tag in this list, the system also provides a probability value. This probability represents the likelihood of describing the image using the specific tag. The Clarifai API can be accessed as a remote web service.

<sup>4</sup><http://www.clarifai.com>



## 1.6.2 Caffe

In several of the proposals of this thesis, we take advantage of the Convolutional Architecture for Fast Feature Embedding (Caffe) [Jia et al., 2014] framework, a fast, modular and well documented DL framework that is widely used by researchers. We chose this framework because of the large community of contributors providing pre-trained models that are ready to be used in any deployment of the framework. The use of the Caffe framework has resulted in solutions to different tasks such as object recognition [Chatfield et al., 2014], or scene classification [Zhou et al., 2014]. This framework is developed and maintained by the Berkeley Vision and Learning Center (BVLC).

### 1.6.2.1 Pre-Trained Caffe Models

In order to train a CNN model, we need to provide both the architecture of the network and the database to be used as a training set. The architecture refers to internal details such as the number of convolutional or fully connected layers, or the spatial operations used in the pooling stages. On the other hand, the training set determines the number of lexical labels used to describe the input image.

From the whole set of available trained models in the Caffe Model Zoo<sup>5</sup>, we selected the seven different candidates that are summarized in Table 1.2. These models differ in the architecture of the CNN used, the dataset used for training them, and the set of predefined lexical labels used by the model. We opted for these models because they were all trained over datasets that consist of images annotated with a large set of generalist lexical labels.

The Caffe framework provides a straightforward way to use pre-trained models released by its vast community. Such models have been trained with different aims, and are defined by the combination of the dataset and the architecture used to generate them.

---

<sup>5</sup><https://github.com/BVLC/caffe/wiki/Model-Zoo>

**Table 1.2:** Details of the seven CNN models evaluated in the proposal

Model Name	CNN ARCH	CL <sup>a</sup>	FCL <sup>b</sup>	Training Datasets	#Labels
ImageNet-AlexNet	AlexNet <sup>c</sup>	5	3	ImageNet2012 <sup>d</sup>	1,000
ImageNet-CaffeNet	AlexNet	5	3	ImageNet2012	1,000
ImageNet-GoogLeNet	GoogLeNet <sup>e</sup>	11	3	ImageNet2012	1,000
ImageNet-VGG	VGG CNN-s <sup>f</sup>	5	3	ImageNet2012	1,000
Hybrid-AlexNet	AlexNet	5	3	Hybrid MIT <sup>g</sup>	1,183
Places-AlexNet	AlexNet	5	3	Places205 MIT <sup>g</sup>	205
Places-GoogLeNet	GoogLeNet	11	3	Places205 MIT	205

<sup>a</sup> Convolution Layers

<sup>b</sup> Fully Connected Layers

<sup>c</sup> [Krizhevsky et al., 2012]

<sup>d</sup> [Russakovsky et al., 2015, Deng et al., 2009]

<sup>e</sup> [Szegedy et al., 2014]

<sup>f</sup> [Chatfield et al., 2014]

<sup>g</sup> [Zhou et al., 2014]

### 1.6.3 CNN architectures for Pre-trained Caffe Models

#### 1.6.3.1 AlexNet

AlexNet architecture [Krizhevsky et al., 2012] was developed for the ImageNet challenge in 2012, producing the first place results of the competition. The architecture follows the basic definition of a CNN, using a successive combination of convolution and pooling layers and a set of fully-connected layers at the end of the design. The network architecture is composed of five convolutional layers and three fully-connected layers. AlexNet models the network output through a Rectifier Linear Units (ReLU) 1.5.1.2, replacing the standard  $\tanh()$  or  $\text{sigmoid}$  function. ReLUs help to improve the training speed of the models.

#### 1.6.3.2 GoogLeNet

GoogLeNet CNN architecture [Szegedy et al., 2014] was presented for the ImageNet recognition challenge in 2014, and it got the first place in the competition and outperformed the results produced by Clarifai in 2013 and AlexNet in 2012. The architecture includes a new concept on the CNN design, namely inception module (Figure 1.15). Inception module works by applying several convolution filters with different sizes ( $1 \times 1$ ,  $3 \times 3$ ,  $5 \times 5$ )

to the same input, at the same moment. Then, the result of each filter is concatenated as the output of the layer. A layer also includes the pooling procedure for the input. This set of small filters allows using smaller parameters, therefore improving the model execution. GoogLeNet consists of 27 layers, where 22 layers have parameters and the five remaining are pooling layers. However, the number of individual blocks used in the design is about 100.

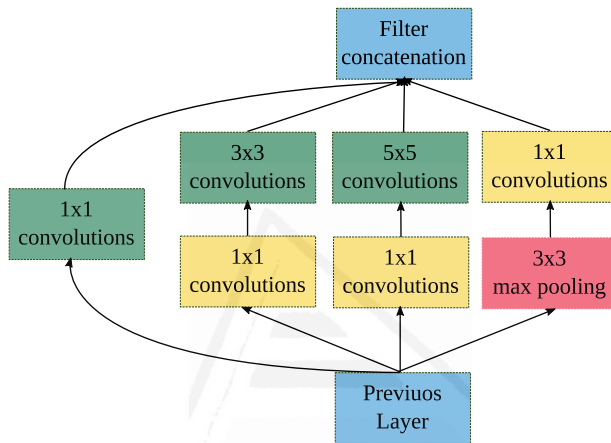


Figure 1.15: Inception Module.

## 1.6.4 Datasets for Pre-trained Caffe Models

### 1.6.4.1 ImageNet 2012 Dataset

ImageNet 2012 dataset [Russakovsky et al., 2015, Deng et al., 2009] is a subset of the original ImageNet dataset which is composed of more than 10 million images from 21,000 categories. The ImageNet 2012 subset contains more than one million images from 1,000 different categories, then for every category it includes approximately 1,000 images. The dataset was created for the ImageNet classification challenge in 2012 and has been used in the competition during the last years. The categories in the dataset include objects of different kind such as: animals, musical instruments, personal care products, among others. ImageNet dataset has been created like an ordered collection of images representing objects and scenes, hierarchically

grouped and categorized. Table 1.3 shows images from four categories in the dataset.

**Table 1.3:** Images from the ImageNet Dataset

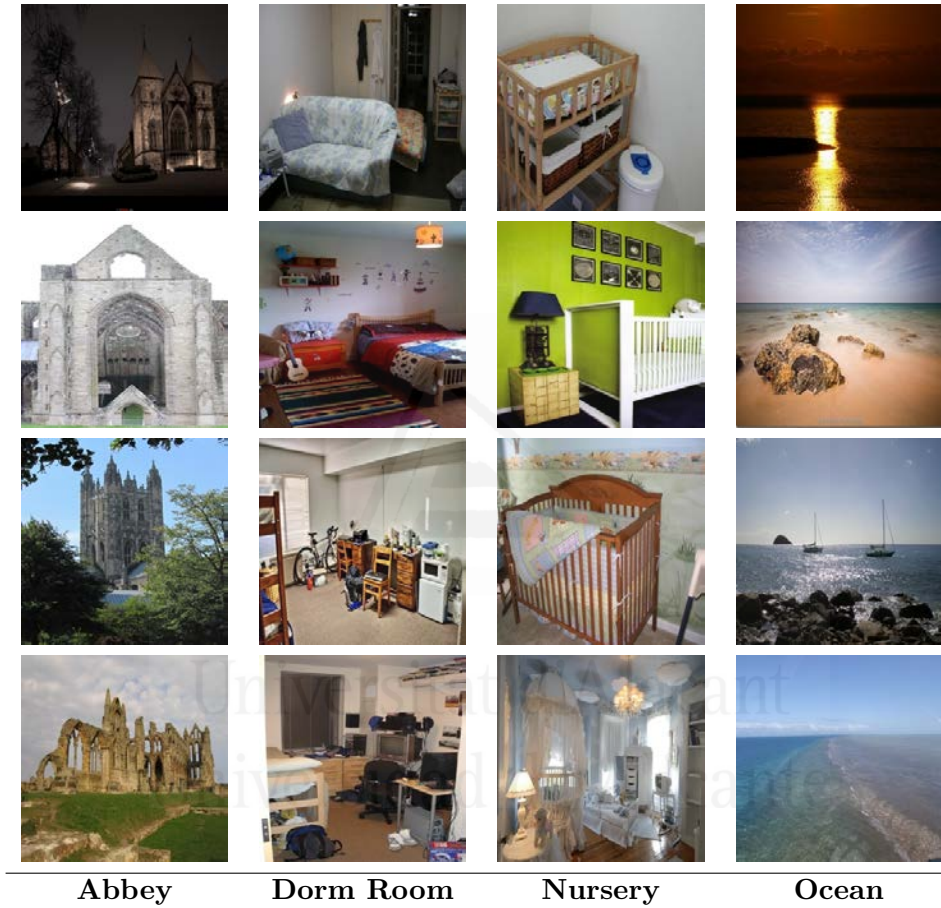
			
			
			
			
<b>Coffe Mug</b>	<b>Hammer</b>	<b>Sax</b>	<b>Tabby</b>

#### 1.6.4.2 Places 205 Dataset

Places 205 [Zhou et al., 2014] dataset consists of a collection of 2,448,873 images that groups 205 scenes categories. This dataset is a subset of the original Places dataset that contains more than seven million images in 476 categories. Places 205 is a scene-centric dataset that focus mainly on the categorization of indoors and outdoors places. The dataset was created by the MIT as a consequence of the lack of scene-centric datasets for scene classification tasks. Moreover, this set was used to learn deep features us-

ing several convolutional neural network architectures to classify images. Table 1.4 shows images from four categories in the dataset.

**Table 1.4:** Images from the Places 205 Dataset



#### 1.6.4.3 Hybrid MIT Dataset

Hybrid dataset [Zhou et al., 2014] was created and presented at the same time as Places dataset and by the same authors. The dataset was constructed by mixing the images belonging to ImageNet 2012 and Places 205 dataset, obtaining 3.5 millions of images. Therefore, categories of the dataset are 1,183, where the duplicate categories names in original datasets were put together. Like Places 205, the dataset was originally built for

learning deep features using CNNs obtaining good results in classification tasks.

## 1.7 Proposal

After describing the motivation of this work, and analyzing the state-of-the-art in Scene Understanding, we have identified a set of approaches in order to develop a robust scene understanding procedure. Among these approaches we have identified an almost unexplored gap in the topic of understanding scenes based on objects present in them. Consequently, we propose to perform an experimental study in this approach aimed at finding a way of fully describing a scene considering the objects lying in the place. As the Scene Understanding task involves object detection and annotation, one of the first steps is to determinate the kind of data to use as input data in our proposal. Evaluating 3D data as an initial footstep, we propose the use of the Growing Neural Gas (GNG) algorithm to represent the input space of noisy point clouds and perform object recognition in the cloud. Based on previous research focusing on neural gases [Garcia-Rodriguez, 2009, Angelopoulou et al., 2005], GNGs have the capacity to grow adapting their topology to represent 2D information, thus producing a smaller representation with a slight noise influence from the input data [Stergiopoulou and Papamarkos, 2006, Doucette et al., 2001, Garcia-Rodriguez et al., 2010, Baena et al., 2013]. Applied to 3D data, the GNG presents a good approach able to tackle noise.

Nevertheless, working with 3D data poses a set of problems such as the lack of a 3D object dataset with enough models to generalize methods to real situations, as well as the computational cost of processing three-dimensional data, which is huge and demands massive storage space. The above-mentioned problems led us to explore new approaches for developing object recognition task. Therefore, based on the positive results that Convolutional Neural Networks (CNNs) have been producing in the latest editions of the ImageNet [Russakovsky et al., 2015] recognition challenge, we propose to carry out an evaluation of the latter as an object detection system. CNNs, initially proposed by [LeCun et al., 1989, LeCun et al.,

1990] in the 90s, are nowadays easily implementable due to hardware optimization. These networks have been tested widely for classification problems involving objects, pedestrians, sound waves, traffic signal, and medical images, among others.

Moreover, an aggregate value of the CNNs is the semantic description capabilities produced by the categories/labels that the network is able to identify and that could be translated as a semantic explanation of the input image. Consequently, we propose to use these semantic labels as a scene descriptor in order to build a supervised scene classification model. On the other hand, we also propose the use of the proposed semantic descriptor for solving topological mapping challenges, and testing the description capabilities of the lexical labels.

Furthermore, semantic descriptors could be suitable for unsupervised places or environment labeling, so we propose their use on this kind of issue in order to achieve a robust scene labeling method. Finally, for tackling the object recognition problem we propose to develop an experimental study for unsupervised object labeling for objects present in a point cloud, and then use it as the training instance of a classifier.

## 1.8 Goals

The main goal of this research is the development and validation of an efficient object-based scene understanding method which will be able to help solve problems related to scene identification for mobile robotics. We seek to analyze state-of-the-art methods in order to find the one that best fits to our goals, as well as to select the kind of data that is more convenient for dealing with the issue.

In order to identify objects with 3D data, we plan to test a GNG representation of objects to deal with noise problems. Regarding 2D information, we will experiment using deep learning techniques, such as CNNs, taking advantage of their description and generalization capabilities. In view of the above points we consider that another primary goal is finding an accurate representation for the scenes by meaning of semantic labels or point cloud features descriptors.

As a secondary goal we will show the benefits of using semantic descriptors generated with pre-trained models for mapping and scene classification problems, as well as using deep learning models in conjunction with 3D features description procedures to build a 3D object classification model that is directly related with the representation goal of this work.

## 1.9 Structure of the dissertation

The PhD dissertation has the following structure:

In Chapter 2 we perform an experimental analysis of several descriptors, keypoint selectors and filter algorithms, in order to determine which one accurately represents an object reducing the noise influence in the detection procedure.

In Chapter 3 we propose the use of the external Deep Learning tool, based on Convolutional Neural Networks Clarifai to represent a scene and use it for scene classification problems.

In Chapter 4 the CNN descriptor is used for building topological maps, based on the semantic annotation produced by deep learning pre-trained models. We compare the results from the Clarifai output with state-of-the-art baseline results.

In Chapter 5 a unsupervised strategy based on hierarchical clustering was proposed to build semantic maps, using CNN descriptors as a representation of the input data.

In Chapter 6 a 3D object classifier is trained using categories assigned by a pre-trained CNN model, in a semi-supervised fashion.

Finally, in Chapter 7, the conclusions extracted from the present work are detailed. Moreover, contributions to the topic are presented and publications derived from this work are enumerated and briefly described. To conclude the chapter, future directions of the research carried out are showed.





Universitat d'Alacant  
Universidad de Alicante

# Object Recognition in Noisy RGB-D Data

---

This chapter presents the evaluation of the GNG to achieve object recognition in noisy scenes, through a feature matching procedure. First, Section 2.1 offers a description of the object recognition problem. Next, in Section 2.2 we review several contributions to the object recognition problem. After that, Section 2.3 introduces and describes the GNG and Voxel Grid methods, that will be used in the experimentation. Then, in Section 2.4 the selected recognition pipeline is explained. Whereas, Section 2.5 describes how the recognition experiments are carried out. After that, in Section 2.6 the results and a discussion of our experiments are showed. Finally, conclusions are discussed in Section 2.7.

## 2.1 Introduction

This chapter describes the procedure carried out to recognize objects in a 3D scene, which is one of the first objectives in order to achieve a robust scene understanding procedure. The experimentation developed in this chapter seeks for evaluating a possible solution to the 3D object recognition problem. Therefore, we will determinate whether employing 3D data solve this obstacle, or will complicate the subsequent work.

3D object recognition is a growing research field which has been stimu-

lated by the well-known advantages offered by the use of 3D sensors compared against 2D based recognition methods. However, there are several difficulties to be overcome in order to achieve effective recognition. Some of these difficulties are: noise, occlusions, rotations, translations, scaling or holes that are present in the raw 3D point clouds provided by current RGB-D sensors such as Microsoft Kinect. Therefore, new algorithms are required to handle these problems when performing a correct object recognition process.

To make the object recognition system robust to noise, we propose the use of a Growing Neural Gas (GNG) [Fritzke, 1995] to represent and reduce the raw point clouds. This self-organizing map learns the distribution of the input space, adapting and preserving its topology. This feature makes it possible to obtain a compact and reduced representation of the input space in a set of 3D neurons and their connections. In addition, we test different keypoint detectors to determine which one obtains better recognition results. This GNG reduction improves the recognition process and reduces noisy 3D values. GNG has been used previously in [Viejo et al., 2012] to filter and reduce point clouds and in [Morell et al., 2014] for 3D map representation. We also compare our proposal against other reduction/filtering methods, such as Voxel Grid. Hence, we present experiments that test a 3D object recognition pipeline with both the raw point cloud, the GNG, and Voxel Grid filtered point clouds.

## 2.2 3D Object Recognition

There are several previous works in the field of 3D object recognition. Some of them provide a survey, review or evaluation of the existing 3D object recognition methods, while other works focus on the proposal of new methods and approaches for the recognition process. In [Guo et al., 2014], a survey of 3D object recognition methods based on local surface features is presented. They divide the recognition process into three basic stages: 3D keypoint detection, feature description, and surface matching. It also describes existing datasets and algorithms used in each stage of the whole process. Other studies, such as [Tombari et al., 2011], focus on the

evaluation of stereo algorithms. This presents an evaluation in terms of the recognition ability of this kind of algorithms. Using a different approach, [Asari et al., 2014] evaluates the different 3D shape descriptors for object recognition to study their feasibility in 3D object recognition.

There are some works that propose novel object recognition pipelines, such as [Hinterstoisser et al., 2011], which combines depth maps and images, achieving good recognition results for heavily cluttered scenes. In [Tombari and Di Stefano, 2010], a novel Hough voting algorithm is proposed to detect free-form shapes in a 3D space, and this also produces good recognition rates. [Pang and Neumann, 2013] describes a general purpose 3D object recognition framework that combines machine learning procedures with 3D local features, without a requirement for a priori object segmentation. This method detects 3D objects in several 3D point cloud scenes, including street and engineering scenes. [Aldoma et al., 2012] proposes a new method called Global Hypothesis Verification (Global HV), which is added to the final phase of the recognition process to discard false positives. Our approach is based on the pipeline presented in this work but, introducing noise into the original point cloud to test the effect of that noise on the recognition process.

## 2.3 3D Filtering methods

In this section, we review some typical methods for representing and reducing 3D data. First, we describe the Growing Neural Gas (GNG) algorithm and how it works. Then, we briefly describe the Voxel Grid method, which is another commonly used data structure, in order to compare our proposed method.

### 2.3.1 Growing Neural Gas (GNG)

A common way to achieve a multi-dimensional reduction is by using self-organizing neural networks where input patterns are projected onto a network of neural units such that similar patterns are projected onto units adjacent in the network and vice versa. As a result of this mapping, a representation of the input patterns is achieved that, in the post-

processing stages, makes it possible to exploit the similarity relations of the input patterns. However, most common approaches are not able to provide good neighborhood and topology preservation if the logical structure of the input pattern is not known a priori. In fact, the most common approaches specify in advance the number of neurons in the network and a graph that represents topological relationships between them, for example a two-dimensional grid, and seek the best match to the given input pattern manifold. When this is not the case, the networks fail to provide good topology preservation, as for example in the case of Kohonen's algorithm [Kohonen, 1995]. The approach presented in this chapter is based on self-organizing networks trained using the Growing Neural Gas learning method [Fritzke, 1995], which is an incremental training algorithm. The links between the neurons in the network are established through competitive Hebbian learning [Martinetz, 1993]. As a result, the algorithm can be used in cases where the topological structure of the input pattern is not known a priori, and yields topology preserving maps of the feature manifold [Martinetz and Schulten, 1994]. The main difference with respect to the original method [Fritzke, 1995] is that in our method a neuron is composed of 3 data elements  $(X, Y, Z)$  representing the point coordinates.

In GNG, nodes in the network compete to determine the set of nodes with the highest similarity to the input distribution. In our case, the input distribution is a finite set of 3D points that can be extracted from different types of sensors. The highest similarity reflects which node, together with its topological neighbors, is the closest to the input sample point, which is the signal generated by the network. The  $n$ -dimensional input signals are randomly generated from a finite input distribution.

The nodes move towards the input distribution by adapting their position to the input data geometry. During the learning process local error measures are gathered to determine where to insert new nodes. New nodes are inserted near the node with the highest accumulated error. At each adaptation step a connection between the winner and its topological neighbors is created as dictated by the competitive Hebbian learning rule. This is continued until an ending condition is fulfilled, as for example evaluation of the optimal network topology, a predefined network size or a deadline.

The network is specified as:

- A set  $N$  of nodes (neurons). Each neuron  $c \in N$  has its associated reference vector  $w_c \in R^d$ . The reference vectors can be considered as positions in the input space of their corresponding neurons.
- A set of edges (connections) between pairs of neurons. These connections are not weighted and their purpose is to define the topological structure. An edge aging scheme is used to remove connections that are invalid due to the motion of the neuron during the adaptation process.

The GNG learning algorithm is presented in Algorithm 2.1, whereas Figure 2.1 shows the working scheme for the GNG method. Using a Growing Neural Gas model to represent 3D data has some advantages over traditional methods such as Voxel Grid. For example, we specify the number of neurons (representative points of the map), while other methods such as Voxel Grid obtain a different number of occupied cells depending on the distribution and resolution of the cells (voxels).

Figure 2.2 shows an example of a GNG representation of one of the objects we use in the experimentation. The GNG forms a map and we use only the neurons as a new, filtered and reduced representation of the original object.

### 2.3.2 Voxel Grid

The Voxel Grid (VG) down-sampling technique is based on the input space sampling using a grid of 3D voxels [Xu et al., 2013]. The VG algorithm defines a grid of 3D cuboids (called voxels) over the input point cloud space and for each voxel (cube), a point is chosen as the representative of all the points that lie on that voxel. It is necessary to define the size of the voxels as this size establishes the resolution of the filtered point cloud, and therefore the number of points that will form the new point cloud. The representative point of each cell is usually the centroid of the voxel's inner points or the center of the voxel grid volume. Thus, a subset of the input space is obtained that roughly represents the underlying surface.

---

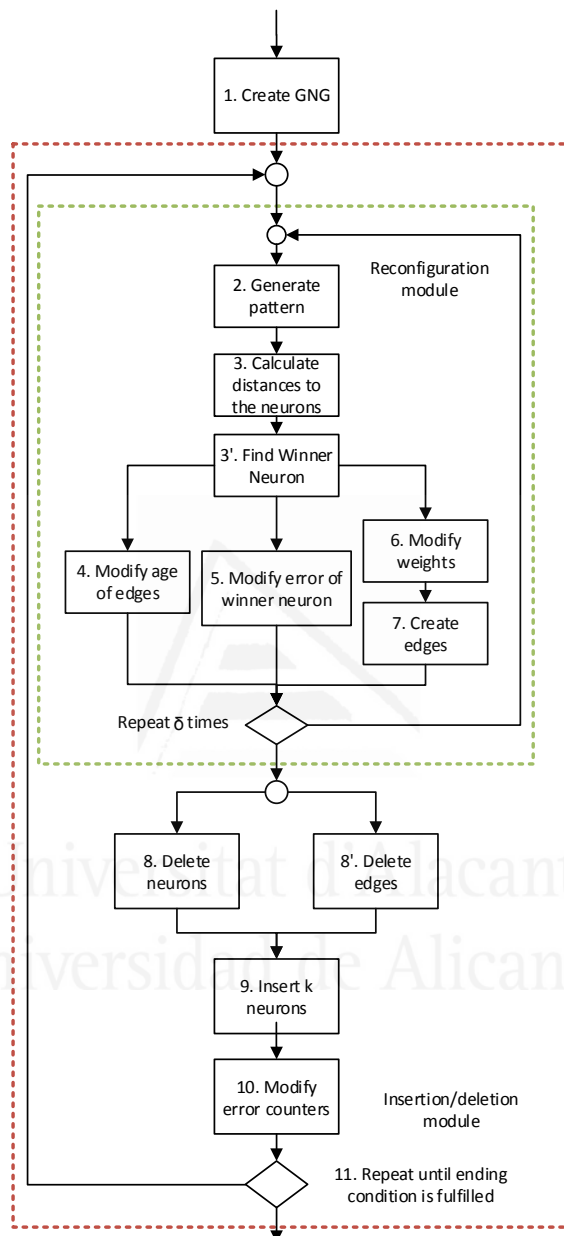
**Algorithm 2.1:** Pseudo-code of the GNG algorithm.
 

---

**input** : N-dimensional input data  
**output**: N-dimensional map

- 1 Start with two neurons  $a$  and  $b$  at random positions  $w_a$  and  $w_b$  in  $R^d$ .
- 2 **repeat**
- 3   **for**  $patterns=0$  **to**  $\delta$  **do**
- 4     Generate at random an input pattern  $\xi$  according to the data distribution  $P(\xi)$  of each input pattern. Find the nearest neuron (winner neuron)  $s_1$  and the second nearest  $s_2$ . Increase the age of all the edges emanating from  $s_1$ . Add the squared distance between the input signal and the winner neuron to a counter error of  $s_1$  such that:
 
$$\Delta error(s_1) = \|w_{s_1} - \xi\|^2 \quad (2.1)$$
- Move the winner neuron  $s_1$  and its topological neighbors (neurons connected to  $s_1$ ) towards  $\xi$  by a learning step  $\epsilon_w$  and  $\epsilon_n$ , respectively, of the total distance:
 
$$\Delta w_{s_1} = \epsilon_w(\xi - w_{s_1}) \quad (2.2)$$
- 5     **forall** *direct neighbors*  $n$  of  $s_1$  **do**
- 6       
$$\Delta w_{s_n} = \epsilon_n(\xi - w_{s_n}) \quad (2.3)$$
- 7     **end forall**
- 8     **if**  $s_1$  and  $s_2$  are connected by an edge **then**
- 9       Set the age of this edge to 0.
- 10    **else**
- 11      Create the connection between  $s_1$  and  $s_2$ .
- 12    **end if**
- 13    Remove the edges larger than  $a_{max}$
- 14    **if** any neuron is isolated (without emanating edges) **then**
- 15      Remove those neurons as well.
- 16    **end if**
- 17    **end for**
- 18    Insert a new neuron as follows:
  - 19      Determine the neuron  $q$  with the maximum accumulated error.
  - 20      Insert a new neuron  $r$  between  $q$  and its furthest neighbor  $f$ :
 
$$w_r = 0.5(w_q + w_f) \quad (2.4)$$
  - Insert new edges connecting the neuron  $r$  with neurons  $q$  and  $f$ , removing the old edge between  $q$  and  $f$ .
- 21    Decrease the error variables of neurons  $q$  and  $f$  multiplying them by a constant  $\alpha$ . Initialize the error variable of  $r$  with the new value of the error variable of  $q$  and  $f$ .
- 22    Decrease all error variables by multiplying them by a constant  $\gamma$ .
- 23 **until** number of neurons reached;

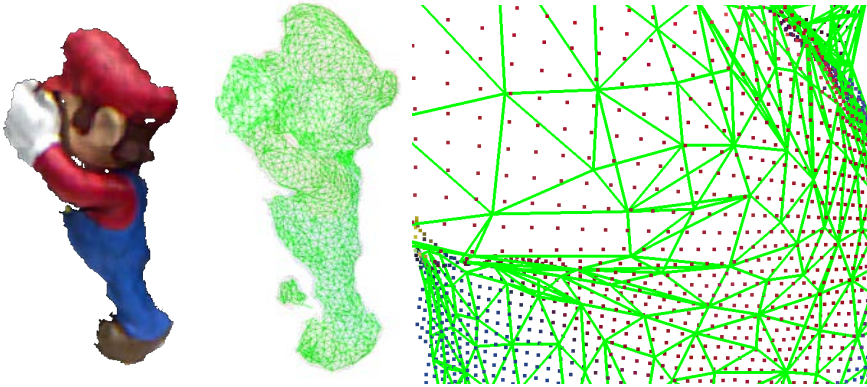
---



**Figure 2.1:** GNG scheme.

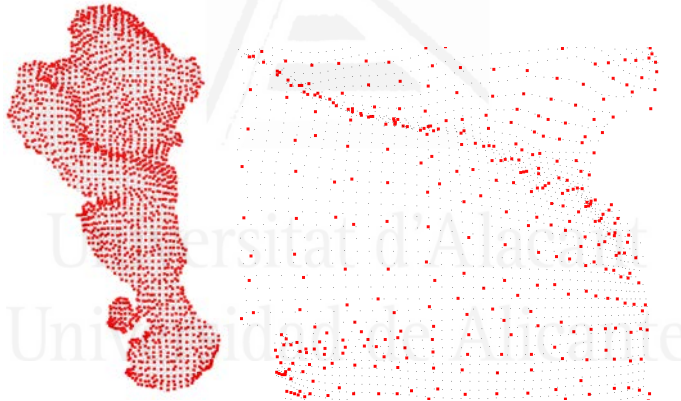
The VG method presents the same problems as other sub-sampling techniques: it is not possible to define the final number of points which represent the surface; geometric information loss due to the reduction of





**Figure 2.2:** Left: Original object. Center: GNG representation of one of the objects used in the experimentation. Right: Zoom of the arm of Mario model.

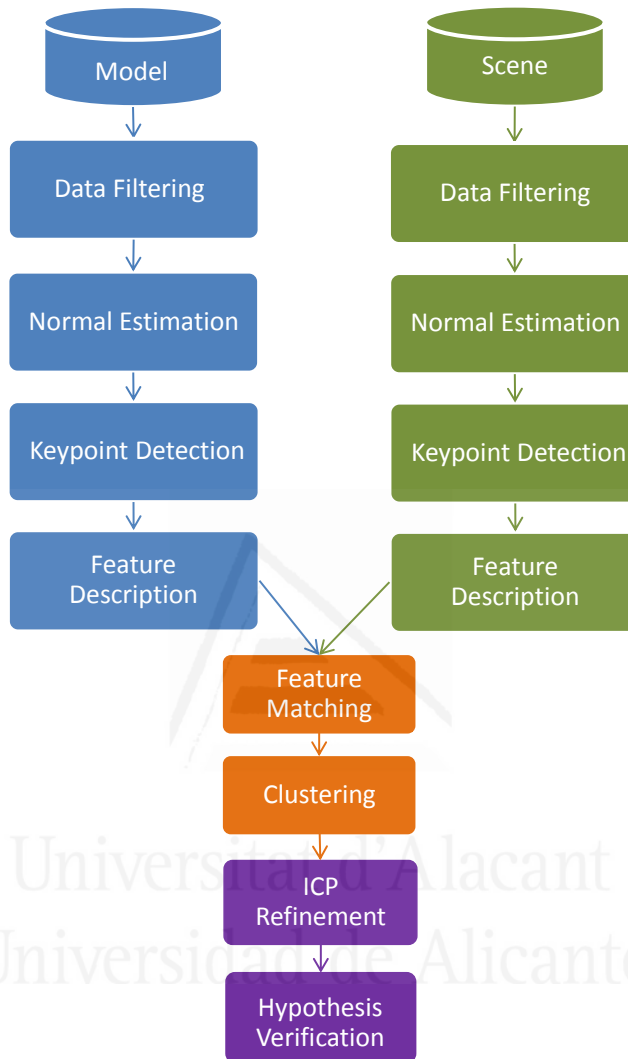
the points inside a voxel; and sensitivity to noisy input spaces.



**Figure 2.3:** Left: Voxel Grid representation of one of the objects we use in the experimentation. Right: Zoom of the arm of Mario.

## 2.4 3D object recognition pipeline

This section describes the overall recognition pipeline, which is based on the one proposed by [Aldoma et al., 2012] (see Figure 2.4). The proposal is based on local features from the point clouds, so the pipeline does not need a segmentation pre-stage. The recognition pipeline is explained in the following subsections.



**Figure 2.4:** Recognition pipeline scheme.

### 2.4.1 Normal extraction

Point normals are frequently used in many areas like computer graphics. Given a geometric surface, it is usually trivial to infer the direction of the normal at a certain point on the surface as the vector perpendicular to the surface at that point. Normals are used to infer the orientation of the surface in a coordinate system. The pipeline will use the normals estimated, both in the scene and the model, to compute other necessary

data such as descriptors and reference frames in the following steps [PCL, 2011].

### 2.4.2 Keypoint detection

After the normals are estimated, the next step is to extract the keypoints for the model and scene point clouds. With this stage we reduce the number of points in the point cloud. It allows us to select only points that are representative of the cloud. This stage makes it possible to reduce the time required to describe the features in further stages. The selection of the keypoints is made using the following keypoint detectors: Uniform Sampling, Harris3D or the Intrinsic Shape Signature(ISS) method.

#### 2.4.2.1 Uniform Sampling

This method builds a 3D grid over the input point cloud. This grid of 3D cuboids, which are called voxels, are located upon the point cloud and only one point is used to represent all the points inside each voxel. This representative point is usually the centroid of the inner points inside a voxel.

#### 2.4.2.2 Harris 3D

Harris 3D[Guo et al., 2014][Sipiran and Bustos, 2011] is a robust point of interest detector for 3D meshes. It adapts the well-known 2D Harris corner detection for images in order to be used for 3D meshes. It has proven to be effective, obtaining high repeatability values. It uses a Gaussian function to smooth the derivative surfaces and mitigate the effect of local deformations introduced by noise, holes, etc. It also proposes an adaptive neighborhood selection which improves feature detection.

#### 2.4.2.3 Intrinsic Shape Signatures(ISS)

The ISS[Zhong, 2009][Guo et al., 2014][Tombari et al., 2013] approach enables both highly discriminative shape matching and efficient pose estimation and registration for 3D point clouds. It is a detector that is carefully crafted to ensure discriminative, descriptive and robust keypoints in

noisy scenes. It is based on the Eigenvalue Decomposition of the scatter matrix of the points belonging to a given neighborhood of a point. This method employs the ratio of two successive eigenvalues ( $\lambda_1, \lambda_2, \lambda_3$ ) to prune the points. Only the points whose ratio between two successive eigenvalues remains below a threshold ( $\tau$ ) are retained. Among the remaining points, the salience is determined by the magnitude of the smallest eigenvalue  $\lambda_1$ ,  $\lambda_2/\lambda_1 < \tau_{21}$  and  $\lambda_3/\lambda_2 < \tau_{32}$ .

### 2.4.3 Feature description

A descriptor codifies the underlying information in a certain neighborhood around a keypoint. Once the keypoints are computed, it is necessary to extract the descriptors for each point cloud. The original work uses the Unique Signatures of Histograms for Local Surface Description (SHOT)[Tombari and Salti, 2011] [Guo et al., 2014][Computer Vision LAB, 2013]. Here we test the pipeline using two other descriptors: Fast Point Feature Histograms (FPFH) [Rusu et al., 2009] and Spin Image [Johnson and Hebert, 1999, Johnson and Hebert, 1998]. We briefly describe them below.

#### 2.4.3.1 Signatures of Histograms for Local Surface Description (SHOT)

This descriptor is an intersection between signatures and histograms. It takes each detected keypoint, builds a local reference frame and then divides the neighborhood space into 3D spherical volumes. Next, according to a function of the angle, between the normal at the keypoint and the near points, a histogram is generated for each spherical volume by accumulating point counts in bins. Joining all the histograms makes the Signature of Histograms of Orientations (SHOT) descriptor. The SHOT descriptor is highly descriptive, computationally efficient and robust to noise.

#### 2.4.3.2 Spin Image

The Spin Image feature descriptor [Johnson and Hebert, 1999, Johnson and Hebert, 1998] is calculated by placing a cylinder at every query point

of the cloud, and orienting the cylinder with the normal at that point. Each cylinder is divided radially and vertically to create a set of volumes. Then, the descriptor is constructed by adding up the neighboring points that are inside each cylinder volume.

### 2.4.3.3 Fast Point Feature Histograms (FPFH)

FPFH [Rusu et al., 2009] is a simplification of the PFH descriptor. To describe a point, the FPFH method computes a set of tuples between the query point and its neighbors. After using the same approach, the tuples are now computed for every selected neighbor and then these values are used to weight a final 16-bin histogram.

### 2.4.4 Feature matching

To determine the correspondences between model and scene descriptors, we used the KDTreeFLANN [Muja and Lowe, 2014] method. This method of the FLANN (Fast Library for Approximate Nearest Neighbors) library uses a kd-tree and an approximate nearest neighbor scheme to find a close feature (the closest is not guaranteed) in a quick way. The structure searches for a scene feature in the model feature set, and only if the distance between the points is less than a threshold, the coordinates are considered as a correspondence and used in the next stage. This structure is commonly used due to its computational time improvement and its fast response results [Marius Muja, 2008].

### 2.4.5 Clustering features using Geometric Consistency

In this step, we group subsets of correspondences found in the above stage into smaller clusters by checking the geometric consistency of pairs of correspondences, using the Geometric Consistency (GC) grouping method. The GC algorithm assumes that the correspondences without geometric consistency will generate large errors in the transformations. Hence, with the geometric consistency method, it is possible to decrease the mismatched correspondences and improve the robustness of the hypothesized

transformations [Guo et al., 2014] [Chen and Bhanu, 2007] [Aldoma et al., 2012].

### 2.4.6 ICP Refinement

This step refines the 6 DoF (Degrees of Freedom) pose by using the Iterative Closest Point (ICP) method. ICP is an iterative method which is able to find the transformation between two point clouds by using a dense representation of the point cloud [Besl and McKay, 1992] [Chen and Medioni, 1991]. ICP uses as initialization the result provided by the clustering stage.

### 2.4.7 Hypothesis Verification

The Hypothesis Verification algorithm was proposed in [Aldoma et al., 2012]. This stage determines whether a given subset of points is a true or false positive. This method takes as input the instances found in the clustering stage and refined by the ICP method. Then, it uses a Simulated Annealing Meta-heuristic algorithm to solve the cost function used to determine whether the hypothesis is valid or not. The Hypothesis Verification method provides a set of instances of the model that match with the scene.

## 2.5 Experimentation

This section describes how the whole set of experiments were organized and performed, as well as, shows the selected dataset for the tests.

### 2.5.1 Dataset

In order to test the different approaches and recognition sceneries, we had selected the SHOT dataset. It consists of a set of cluttered scenes and objects point clouds, for more details of this dataset see Section 1.4.1.

### 2.5.2 Experimentation setup

The experiment consists in searching a selected model in a scene using the recognition pipeline described above. As we are testing the method with different noise levels, we have applied five different levels of Gaussian noise with 0.001, 0.0025, 0.005, 0.0075 and 0.01 meters of standard deviation. We only applied noisy values in the scene point clouds as the stored models are supposed to have a higher quality level. These new clouds with noise, named the RAWNoise dataset, are reduced or filtered using the GNG and VoxelGrid methods.

We apply the GNG algorithm to reduce the scene clouds to 10,000, 15,000, 17,500 and 20,000 representative points, which we call the GNG-Noise dataset. This process is repeated using the Voxel Grid method to achieve a similar reduction and obtaining the VoxelGridNoise dataset. We repeat this process with each keypoint detector and descriptor. For the point clouds in the dataset, the GNG algorithm has been executed using the following parameters:  $\delta = 2,000$ ,  $\epsilon_w = 0.1$ ,  $\epsilon_n = 0.001$ ,  $\alpha = 0.5$  and  $\alpha_{max} = 250$ . These produced good results in our previous works with GNG and 3D data.

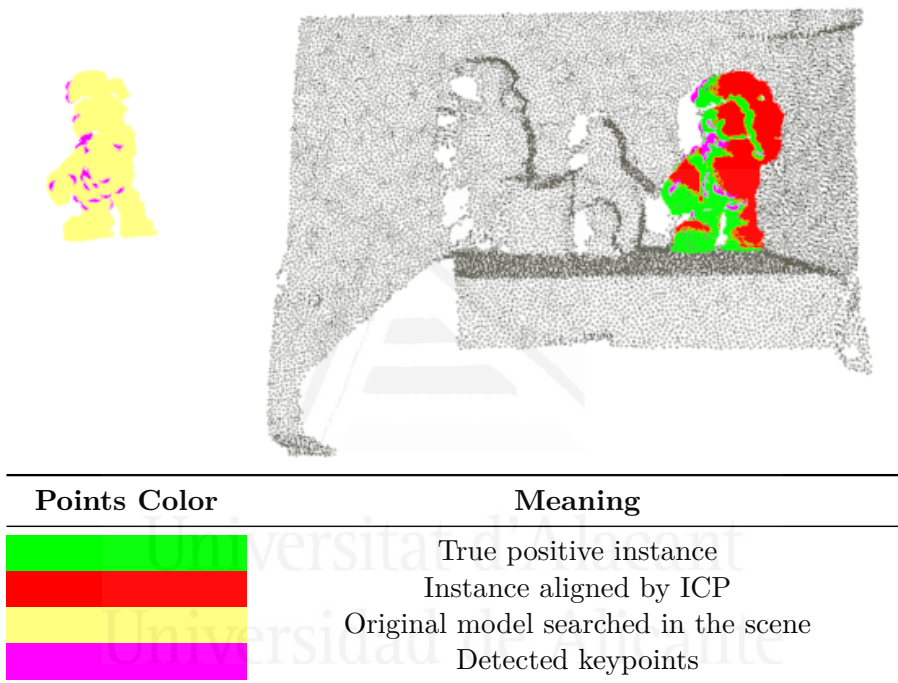
Finally, we have a dataset with six different sets of point clouds and five levels of noise for the scene datasets. We decided to test every combination of model and scene on the different datasets to obtain the most representative values. Table 2.1 shows all the possible combinations of our datasets. The first word of each pair is the method applied to the model, and the second one is the applied to the scene. We tested the system with the 49 recognition instances available in the dataset.

**Table 2.1:** List of the experiment combinations.

<b>GNG Models</b>	<b>RAW Models</b>	<b>Voxel Grid Models</b>
GNG_GNG	RAW_GNG	VoxelGrid_GNG
GNG_GNGNoise	RAW_GNGNoise	VoxelGrid_GNGNoise
GNG_RAW	RAW_RAW	VoxelGrid_RAW
GNG_RAWNoise	RAW_RAWNoise	VoxelGrid_RAWNoise
GNG_VoxelGrid	RAW_VoxelGrid	VoxelGrid_VoxelGrid
GNG_VoxelGridNoise	RAW_VoxelGridNoise	VoxelGrid_VoxelGridNoise

To measure the performance of the recognition pipeline, we use the

Hypothesis Verification algorithm, which analyzes the results and provides us with the true positives of the recognition method over the different datasets. When the system finds a true positive, it only takes the instance with the most matched points between the model and the scene that has been located, and shows a screen with the model superimposed on the scene, in the position where the instance has been located (see Figure 2.5).



**Figure 2.5:** Recognition result obtained by the pipeline.

## 2.6 Results

This section presents the results obtained after the execution of the different sets of experiments. Results are presented for each descriptor and, then, for each keypoint detector.



### 2.6.1 Results for SHOT descriptor

Table 2.2 shows the percentage of true positives obtained using the Uniform Sampling keypoint detector with the SHOT descriptor. The number in the green shade represents the higher percentages, while yellow indicates the middle values and red the lowest ones. The recognition percentage obtained for a non-filtered scene and model was 86%, where we used all the points in the clouds, obtaining the highest recognition value for all the experiments using Uniform Sampling. With this keypoint detector the better recognition results when adding noise were those filtered with the GNG method. In the presence of noise, the highest recognition rate obtained was 83%, using the scenes with the 0.005 noise level and filtered using the GNG method with 10,000 representative points. On the other hand, the results for a raw point cloud with the same noise level do not provide true positive recognition results. Comparing with the value obtained using scenes filtered with VG, GNG maintains the recognition rates. It is noteworthy that with a 0.01 noise level we did not obtain true positive recognition results in any experiment.

**Table 2.2:** Results for the experiments using SHOT and Uniform Sampling.

Models	Scene's Noise	Scenes								
		RAW	GNG				VoxelGrid			
		All Points	10000	15000	17500	20000	10000	15000	17500	20000
RAW	0	86	50	47	53	65	9	31	23	31
	0.001	82	45	49	50	57	17	23	27	28
	0.0025	20	58	67	74	73	9	16	22	36
	0.005	0	75	77	69	73	0	0	0	0
	0.0075	0	37	43	31	22	0	0	0	0
	0.01	0	3	0	0	0	0	0	0	0
GNG	0	84	52	58	57	70	23	29	26	42
	0.001	69	59	64	67	72	18	38	40	49
	0.0025	9	59	67	79	73	32	24	22	40
	0.005	0	83	71	73	76	0	0	3	0
	0.0075	0	49	31	14	10	0	0	0	0
	0.01	0	0	0	0	0	0	0	0	0
VoxelGrid	0	78	51	66	64	70	27	38	42	47
	0.001	65	48	64	62	70	27	38	37	45
	0.0025	0	63	74	76	66	24	35	35	41
	0.005	0	60	74	63	69	0	0	3	8
	0.0075	0	36	24	26	13	0	0	0	0
	0.01	0	0	0	0	0	0	0	0	0

Table 2.3 shows the results for the Harris 3D keypoint detector. Using this detector the highest recognition rate was 81%, with a 0.005 noise level and filtered using the GNG method with 15,000 representative points.

For non-filtered scenes and models, the recognition percentage for this detector was 84%. The noisy scenes filtered by GNG obtain better results outperforming those filtered using the Voxel Grid method. Again, with this detector no true positive results were obtained for the cloud with a 0.01 noise level.

**Table 2.3:** Results for the experiments using SHOT and Harris 3D.

Models	Scene's Noise	Scenes								
		RAW	GNG				VoxelGrid			
		All Points	10000	15000	17500	20000	10000	15000	17500	20000
RAW	0	84	52	58	36	43	3	22	12	17
	0.001	80	49	51	48	51	3	14	15	23
	0.0025	0	63	57	54	74	9	21	31	44
	0.005	0	68	77	77	76	0	0	3	6
	0.0075	0	28	33	19	13	0	0	0	0
	0.01	0	0	0	0	0	0	0	0	0
GNG	0	76	77	71	55	76	15	39	26	41
	0.001	58	64	66	67	68	19	32	40	50
	0.0025	0	69	77	70	77	21	34	53	49
	0.005	0	73	81	78	71	0	0	5	0
	0.0075	0	43	24	15	8	0	4	4	0
	0.01	0	0	0	0	0	0	0	0	0
VoxelGrid	0	59	69	65	52	67	21	40	40	54
	0.001	42	64	64	66	72	19	52	47	50
	0.0025	0	64	63	67	70	15	43	65	64
	0.005	0	69	77	67	61	0	0	0	3
	0.0075	0	22	18	5	3	0	0	0	0
	0.01	0	0	0	0	6	0	0	0	0

In Table 2.4 we show the results for the ISS keypoint detector. Using this detector the highest recognition value obtained was 74% , using a 0.001 noise level and filtered with GNG with 20,000 representatives. Non-filtered scenes and models obtained 86% recognition, and for the scenes with a 0.01 noise level we did not obtain any true positive result.

Table 2.5 shows the mean of the recognition results obtained for the keypoint detectors evaluated. The highest values were obtained when using the scenes filtered by the GNG method. For the noisy clouds, the highest value obtained was 72% for the cloud filtered with the GNG method with 20,000 representatives and a 0.001 noise level. GNG always obtains better results than Voxel Grid, and the number of representatives is an influential factor for recognition. With the GNG filtered scenes we are able to recognize objects with a higher noise level.

Another noteworthy result is that GNG achieved a better rate than that obtained for the point clouds filtered with the Voxel Grid method. Table 2.6 shows the mean of all the recognition results, grouped by the

**Table 2.4:** Results for the experiments using SHOT and Intrinsic Shape Signature.

Models	Scene's Noise	Scenes									
		RAW	GNG				VoxelGrid				
		All Points	10000	15000	17500	20000	10000	15000	17500	20000	
RAW	0	86	32	39	55	50	0	9	13	33	
	0.001	88	44	55	41	44	0	22	25	28	
	0.0025	13	36	46	59	64	0	11	24	33	
	0.005	0	29	56	44	50	0	0	0	0	
	0.0075	0	8	3	0	0	0	0	0	0	
	0.01	0	0	0	0	0	0	0	0	0	
GNG	0	80	28	51	56	56	7	20	24	31	
	0.001	61	36	49	55	62	5	30	30	42	
	0.0025	0	40	48	56	61	0	14	26	36	
	0.005	0	24	48	39	34	0	0	0	0	
	0.0075	0	6	0	0	0	0	0	0	0	
	0.01	0	0	0	0	0	0	0	0	0	
VoxelGrid	0	80	26	47	50	63	22	47	39	43	
	0.001	69	34	44	65	74	19	43	49	47	
	0.0025	0	30	44	66	67	17	33	38	49	
	0.005	0	22	39	39	33	0	0	0	0	
	0.0075	0	0	4	0	0	0	0	0	0	
	0.01	0	0	0	0	0	0	0	0	0	

**Table 2.5:** Mean values of the results for the experiments using SHOT.

Models	Scene's Noise	Scenes									
		RAW	GNG				VoxelGrid				
		All Points	10000	15000	17500	20000	10000	15000	17500	20000	
RAW	0	85	45	48	48	53	4	21	16	27	
	0.001	83	46	52	46	51	7	20	22	26	
	0.0025	11	52	57	63	70	6	16	26	38	
	0.005	0	57	70	63	66	0	0	1	2	
	0.0075	0	24	26	17	12	0	0	0	0	
	0.01	0	1	0	0	0	0	0	0	0	
GNG	0	80	52	60	56	67	15	29	25	38	
	0.001	63	53	60	63	67	14	33	37	47	
	0.0025	3	56	64	68	70	18	24	34	42	
	0.005	0	60	67	63	60	0	0	3	0	
	0.0075	0	32	19	10	6	0	1	1	0	
	0.01	0	0	0	0	0	0	0	0	0	
VoxelGrid	0	72	49	59	56	67	23	42	40	48	
	0.001	59	49	57	64	72	22	45	44	47	
	0.0025	0	53	60	70	68	19	37	46	52	
	0.005	0	50	63	56	55	0	0	1	4	
	0.0075	0	19	16	10	5	0	0	0	0	
	0.01	0	0	0	0	2	0	0	0	0	

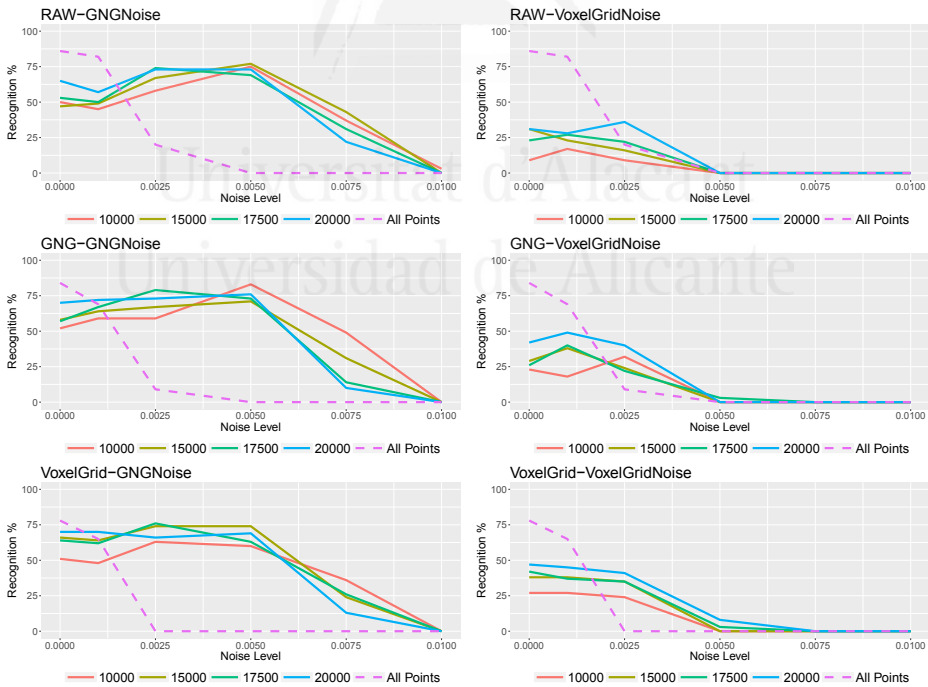
numbers of representative points. The left part in this table represents the values achieved by the pipeline using the GNG filtered clouds and the right part shows the values obtained using the Voxel Grid method. In this table, we see that using the point clouds filtered by the GNG with 20,000 representatives obtains better results than the one filtered one with fewer representatives.

Figure 2.6 shows the effect of the noise on the recognition process. The dotted line represents the experiments using the RawNoise scenes with the

**Table 2.6:** Mean values of the results for the experiments using the evaluated keypoint detectors with the SHOT descriptor, grouped by the number of representatives.

	Scenes								
	RAW	GNG				VoxelGrid			
	All Points	10000	15000	17500	20000	10000	15000	17500	20000
RAW	30	38	42	39	42	3	9	11	16
GNG	24	42	45	43	45	8	15	17	21
VoxelGrid	22	37	43	43	45	11	21	22	25

models indicated in the chart. Using a raw point cloud scene (noise level 0), recognition reaches the highest values but, adding different noise levels reduces this effectiveness even with low noise levels. These charts show how GNG outperforms the Voxel Grid method and helps to maintain the recognition process when noisy clouds are used. These results indicate that the algorithm only tolerates noise levels up to 0.0075 meters of standard deviation.



**Figure 2.6:** Charts representing the mean results for different experiment sets using the SHOT descriptor.

Calculating the mean of the recognition rates grouped by filtering method, Uniform Sampling obtains the highest true positive recognition rate with 47,8%, followed by Harris 3D with 46,8% and, finally, ISS with 31,2%.

### 2.6.2 Results for the FPFH feature descriptor

Tables 2.7, 2.8 and 2.9 show the results for Uniform Sampling, Harris3D and ISS, respectively. Table 2.10 shows the mean values grouped by the number of representatives. Finally, Figure 2.7 shows a comparison of the mean rate recognition results.

With this descriptor, although using the raw data the recognition rate is 92%, while in clouds with low noise the recognition rate decreases quickly. However, GNG is able to provide good recognition rates for noise levels up to 0.0025. For GNG, the recognition rate is better than the one provided by the SHOT descriptor. This behavior is the same for the three keypoint detectors. Regarding the mean values, GNG yields better results, although the different numbers of neurons.

Comparing the results obtained for each detector we can see that Harris 3D achieves the highest true positive mean recognition rate results for GNG, with 40,2%, Uniform Sampling obtained 39,8%, and ISS 28,8%.

**Table 2.7:** Results for the experiments using FPFH and Uniform Sampling.

Models	Scene's Noise	Scenes								
		RAW	GNG				VoxelGrid			
		All Points	10000	15000	17500	20000	10000	15000	17500	20000
RAW	0	92	45	73	67	76	22	57	61	63
	0.001	24	55	61	71	73	33	41	47	51
	0.0025	2	43	65	84	80	33	35	39	37
	0.005	2	47	51	37	18	4	4	0	2
	0.0075	2	12	2	6	8	4	8	2	2
	0.01	0	0	2	0	2	4	0	0	0
GNG	0	80	47	80	73	80	39	53	61	63
	0.001	14	53	61	63	76	51	59	69	59
	0.0025	0	65	63	69	65	39	37	31	29
	0.005	0	43	29	24	18	0	6	2	2
	0.0075	0	2	0	2	0	2	0	2	2
	0.01	8	0	0	0	0	4	2	4	0
VoxelGrid	0	80	53	78	71	82	31	67	57	65
	0.001	18	41	69	78	76	47	59	61	71
	0.0025	0	47	65	67	82	27	39	39	47
	0.005	0	45	47	29	22	8	0	0	0
	0.0075	0	14	0	2	4	0	0	0	0
	0.01	6	2	0	0	0	2	0	0	2

**Table 2.8:** Results for the experiments using FPFH and Harris 3D.

Models	Scene's Noise	Scenes									
		RAW	GNG				VoxelGrid				
		All Points	10000	15000	17500	20000	10000	15000	17500	20000	
RAW	0	90	57	65	65	65	33	35	37	45	
	0.001	27	61	53	65	65	27	39	53	53	
	0.0025	0	63	71	69	78	35	31	31	22	
	0.005	0	61	51	43	18	8	4	6	2	
	0.0075	2	10	10	4	4	2	2	2	2	
	0.01	6	2	0	0	0	4	0	4	2	
GNG	0	76	61	61	76	69	37	35	45	53	
	0.001	14	55	67	67	65	43	45	43	43	
	0.0025	2	67	73	67	69	31	24	31	14	
	0.005	2	55	41	20	16	2	2	2	6	
	0.0075	2	2	8	0	2	4	2	0	0	
	0.01	6	2	2	0	4	0	0	6	0	
VoxelGrid	0	73	55	78	63	63	31	63	47	55	
	0.001	16	51	57	69	61	33	51	47	51	
	0.0025	0	61	61	63	63	24	33	39	37	
	0.005	6	59	45	45	20	2	6	4	8	
	0.0075	2	12	10	8	4	4	0	2	2	
	0.01	2	4	0	2	2	4	4	2	0	

**Table 2.9:** Results for the experiments using FPFH and Intrinsic Shape Signature.

Models	Scene's Noise	Scenes									
		RAW	GNG				VoxelGrid				
		All Points	10000	15000	17500	20000	10000	15000	17500	20000	
RAW	0	92	16	31	51	57	16	39	35	53	
	0.001	61	18	43	55	61	22	39	51	55	
	0.0025	20	27	65	61	65	20	31	37	31	
	0.005	8	29	29	18	20	0	2	2	0	
	0.0075	2	6	0	2	2	0	0	0	0	
	0.01	8	0	0	0	2	0	0	0	0	
GNG	0	84	33	57	57	63	24	55	43	51	
	0.001	49	35	61	55	73	22	45	51	57	
	0.0025	10	41	55	63	51	22	16	24	22	
	0.005	4	31	31	18	4	2	0	2	0	
	0.0075	0	2	0	4	0	0	0	0	0	
	0.01	0	0	0	0	0	0	0	0	0	
VoxelGrid	0	92	33	49	51	67	24	57	41	61	
	0.001	57	27	49	57	63	33	47	53	63	
	0.0025	12	29	53	55	69	16	33	35	41	
	0.005	4	18	20	22	16	0	0	0	0	
	0.0075	0	2	0	0	0	0	0	0	0	
	0.01	0	0	0	0	0	0	0	0	0	

### 2.6.3 Results for the Spin Image feature descriptor

Tables 2.12, 2.13 and 2.14 show the results for the experiments with the different keypoints using the Spin Image feature descriptor. We also have the same tables as the previous descriptors for the mean values (Table 2.15) and the mean values grouped by the number of representatives (Table 2.16). Figure 2.8 shows a comparison of the mean rate recognition results.

For this set of experiments, GNG achieves better results than the Voxel Grid filtering method, but the recognition rates were very low. The best

**Table 2.10:** Mean values of the results for the experiments using FPFH.

Models	Scene's Noise	Scenes								
		RAW All Points	GNG				VoxelGrid			
			10000	15000	17500	20000	10000	15000	17500	20000
RAW	0	91	39	56	61	66	24	44	44	54
	0.001	37	45	52	64	67	27	39	50	53
	0.0025	7	44	67	71	74	29	32	35	30
	0.005	3	46	44	33	19	4	3	3	1
	0.0075	2	10	4	4	5	2	3	1	1
	0.01	5	1	1	0	1	3	0	1	1
GNG	0	80	47	66	69	71	33	48	50	56
	0.001	26	48	63	62	71	39	50	54	53
	0.0025	4	58	64	67	62	31	26	29	22
	0.005	2	43	33	21	13	1	3	2	3
	0.0075	1	2	3	2	1	2	1	1	1
	0.01	5	1	1	0	1	1	1	3	0
VoxelGrid	0	82	47	68	62	71	29	63	48	61
	0.001	31	39	59	68	67	37	52	54	62
	0.0025	4	46	60	62	71	22	35	37	41
	0.005	3	41	37	32	20	3	2	1	3
	0.0075	1	10	3	3	3	1	0	1	1
	0.01	3	2	0	1	1	2	1	1	1

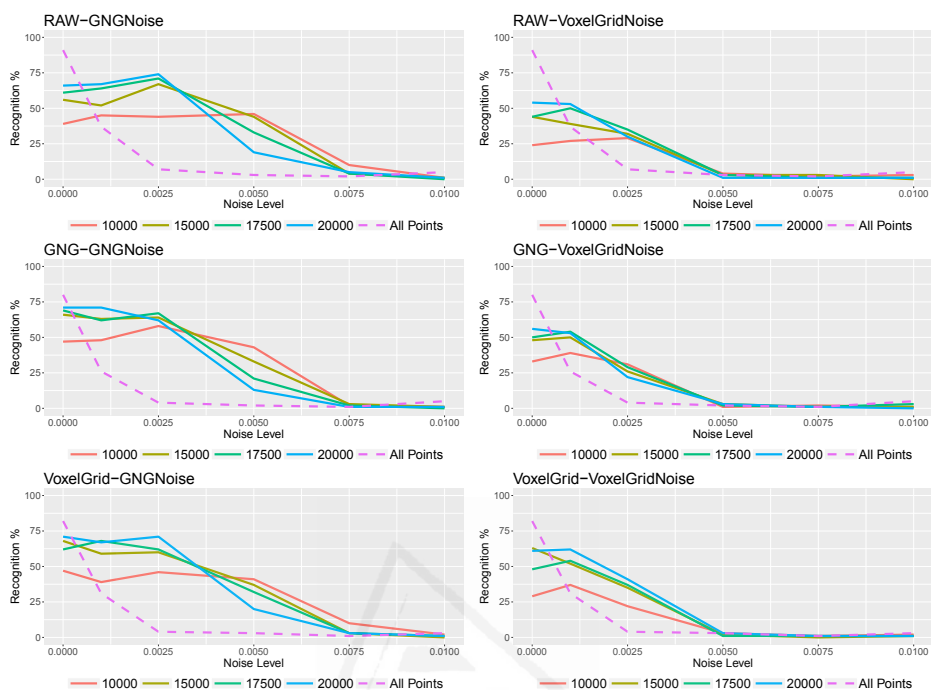
**Table 2.11:** Mean values of the results for the experiments using the evaluated keypoint detectors with the FPFH descriptor, grouped by the number of representatives.

	Scenes								
	RAW All Points	GNG				VoxelGrid			
		10000	15000	17500	20000	10000	15000	17500	20000
RAW	24	31	37	39	39	15	20	23	23
GNG	20	33	38	37	37	18	21	23	22
VoxelGrid	21	31	38	38	39	16	26	24	28

recognition rate was obtained by the ISS detector and the Spin Image descriptor. This indicates that the use of a filter method with the Spin Image descriptor is not appropriate. This is due to the fact that the Spin Image uses a number of the neighbors in a given region around the keypoint to build the descriptor. With the point reduction performed by GNG or Voxel Grid, the descriptor obtained could not be good enough for recognition tasks. Furthermore, this combination of detector and descriptor is computationally expensive, comparing it with Uniform Sampling and SHOT.

## 2.6.4 Discussion

These results support our proposal that the use of GNG improves the results of recognition in noisy point clouds. Comparing the results obtained with the descriptors, the highest recognition percentage for the GNG point



**Figure 2.7:** Charts representing the mean results for experiment sets using the FPFH descriptor.

**Table 2.12:** Results for the experiments using Spin Image and Uniform Sampling.

Models	Scene's Noise	Scenes											
		RAW All Points	GNG				VoxelGrid						
			10000	15000	17500	20000	10000	15000	17500	20000			
RAW	0	24	10	14	6	6	8	2	2	6			
	0.001	16	8	8	10	8	0	0	12	4			
	0.0025	6	2	14	16	10	2	0	4	4			
	0.005	0	8	2	6	12	4	4	4	10			
	0.0075	14	8	2	10	14	8	8	4	0			
	0.01	41	4	2	10	14	14	10	8	12			
GNG	0	18	6	12	8	14	6	2	0	4			
	0.001	10	6	12	12	10	0	2	8	2			
	0.0025	14	14	10	14	14	2	10	8	10			
	0.005	6	2	4	8	12	2	2	2	4			
	0.0075	47	6	8	4	14	8	6	10	8			
	0.01	57	8	2	8	8	12	10	14	33			
VoxelGrid	0	24	6	6	8	6	2	8	2	4			
	0.001	10	6	10	16	14	2	8	6	2			
	0.0025	10	10	14	8	8	2	4	2	4			
	0.005	2	8	8	6	14	8	4	8	4			
	0.0075	31	4	4	4	4	4	8	8	10			
	0.01	49	2	4	6	6	0	4	18	20			

clouds was achieved by the SHOT feature descriptor with 41,9%, followed by FPFH with 36,2% and, finally, Spin Image with 7,2%.

Table 2.17 and Figure 2.9 show a comparison of the different detectors,



**Table 2.13:** Results for the experiments using Spin Image and Harris 3D.

Models	Scene's Noise	Scenes									
		RAW	GNG				VoxelGrid				
		All Points	10000	15000	17500	20000	10000	15000	17500	20000	
RAW	0	12	10	4	2	8	8	4	8	0	
	0.001	12	4	2	6	12	12	0	8	0	
	0.0025	6	6	18	8	4	14	6	6	10	
	0.005	6	22	10	29	16	2	6	16	14	
	0.0075	18	12	12	18	31	6	2	4	4	
	0.01	51	12	14	10	18	6	6	12	14	
GNG	0	10	10	16	12	10	10	8	6	6	
	0.001	6	10	6	10	2	4	8	6	6	
	0.0025	2	4	6	10	6	10	10	14	8	
	0.005	2	10	10	14	18	6	2	10	14	
	0.0075	16	2	14	12	20	8	2	4	18	
	0.01	71	6	4	12	12	14	18	29	27	
VoxelGrid	0	6	24	12	2	8	4	6	6	4	
	0.001	4	10	8	10	6	4	4	8	6	
	0.0025	2	10	6	4	4	8	6	6	4	
	0.005	2	10	18	24	10	6	4	8	12	
	0.0075	20	12	12	18	24	10	6	6	18	
	0.01	57	2	6	6	10	4	8	6	27	

**Table 2.14:** Results for the experiments using Spin Image and Intrinsic Shape Signature.

Models	Scene's Noise	Scenes									
		RAW	GNG				VoxelGrid				
		All Points	10000	15000	17500	20000	10000	15000	17500	20000	
RAW	0	63	2	0	2	6	2	0	2	0	
	0.001	71	2	0	4	2	2	8	8	6	
	0.0025	39	0	0	6	2	2	0	4	6	
	0.005	6	0	2	6	2	0	6	4	2	
	0.0075	10	0	0	4	6	2	2	2	0	
	0.01	47	2	2	6	0	2	2	0	2	
GNG	0	37	0	2	8	2	0	0	0	4	
	0.001	53	0	6	2	2	2	2	0	4	
	0.0025	29	0	4	8	0	0	2	4	8	
	0.005	4	0	4	2	4	0	0	0	4	
	0.0075	51	0	2	2	2	0	0	2	0	
	0.01	82	0	0	0	0	2	2	0	0	
VoxelGrid	0	27	2	4	4	2	12	4	2	4	
	0.001	63	0	2	6	4	0	6	0	6	
	0.0025	35	0	0	6	6	4	2	4	16	
	0.005	18	0	2	4	2	0	2	4	2	
	0.0075	55	0	0	2	0	0	0	2	0	
	0.01	82	0	0	0	0	0	2	0	0	

descriptors and filter methods. The best recognition rate is achieved by the combination of Uniform Sampling and SHOT, using GNG with 17,500 neurons.

The Spin Image descriptor cannot be used with a noise reduction method due to its way of building the descriptor. Furthermore, this descriptor is computationally expensive.

In SHOT and FPFH with GNG and in Spin Image with the raw data, the recognition rate is lower for clouds without noise than with noise. This is due to the way the descriptor is calculated. For SHOT and FPFH

**Table 2.15:** Mean values of the results for the experiments using Spin Image.

Models	Scene's Noise	Scenes									
		RAW	GNG				VoxelGrid				
		All Points	10000	15000	17500	20000	10000	15000	17500	20000	
RAW	0	33	7	6	3	7	6	2	4	2	
	0.001	33	5	3	7	7	5	3	10	3	
	0.0025	17	3	11	10	5	6	2	5	7	
	0.005	4	10	5	14	10	2	5	8	9	
	0.0075	14	7	5	11	17	5	4	3	1	
	0.01	46	6	6	9	11	7	6	7	10	
GNG	0	22	5	10	10	9	5	3	2	5	
	0.001	23	5	8	8	5	2	4	5	4	
	0.0025	15	6	7	11	7	4	7	9	9	
	0.005	4	4	6	8	12	3	1	4	7	
	0.0075	38	3	8	6	12	5	3	5	9	
	0.01	70	5	2	7	7	10	10	14	20	
VoxelGrid	0	19	11	7	5	5	6	6	3	4	
	0.001	26	5	7	11	8	2	6	5	5	
	0.0025	16	7	7	6	6	5	4	7	8	
	0.005	7	6	10	12	9	5	3	4	6	
	0.0075	35	5	5	8	10	5	5	5	10	
	0.01	63	1	3	4	5	1	5	8	16	

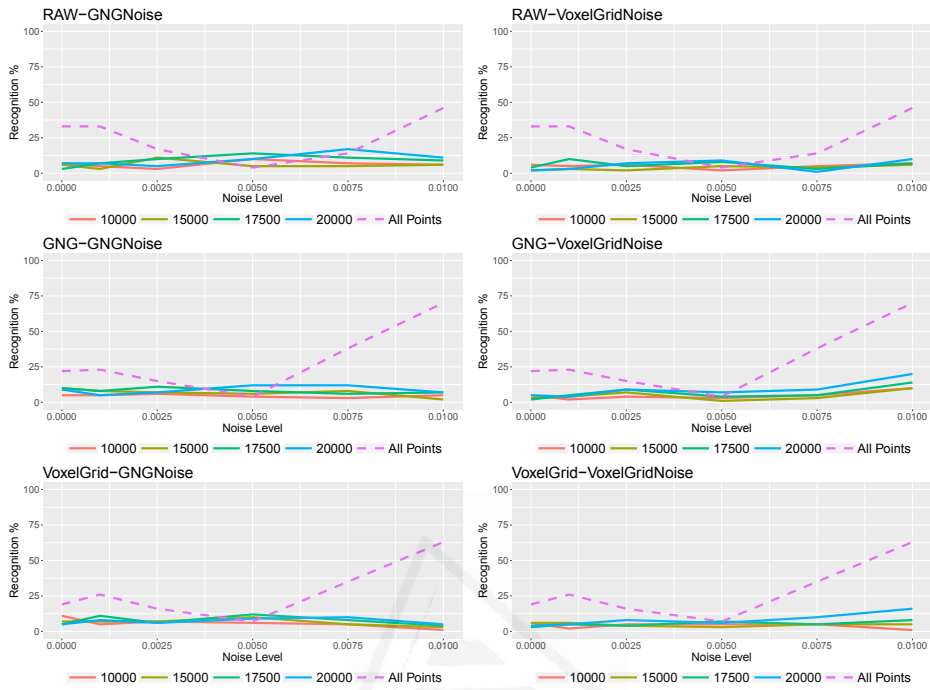
**Table 2.16:** Mean values of the results for the experiments using the evaluated keypoint detectors with the Spin Image descriptor, grouped by the number of representatives.

	Scenes								
	RAW	GNG				VoxelGrid			
	All Points	10000	15000	17500	20000	10000	15000	17500	20000
RAW	25	6	6	9	10	5	4	6	5
GNG	29	5	7	8	9	5	5	7	9
VoxelGrid	28	6	7	8	7	4	5	5	8

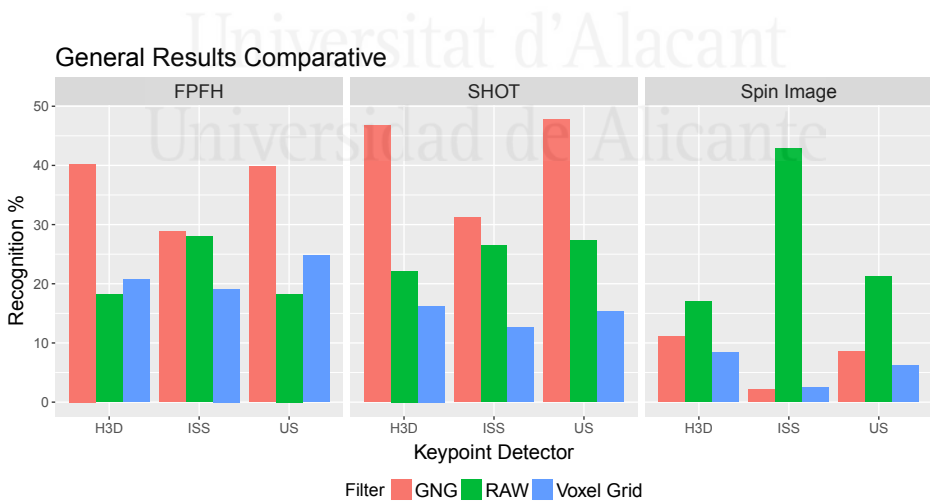
**Table 2.17:** Results for the experiments by detector, descriptor and filter.

Feature Descriptor	Keypoint Detector	Filter		
		RAW	GNG	VG
Spin Image	US	21.20	8.53	6.18
	H3D	17.01	11.03	8.42
	ISS	42.86	2.15	2.44
SHOT	US	27.35	47.79	15.35
	H3D	22.14	46.78	16.25
	ISS	26.47	31.22	12.64
FPFH	US	18.25	39.82	24.83
	H3D	18.14	40.22	20.75
	ISS	28.00	28.83	19.10

with the raw data, the behavior is as the expected: the more noise there is, the lower the recognition rate becomes. But GNG is able to capture the topology of the 3D data and this topology is well adapted to both descriptors. However, for Spin Image and the raw data, the descriptor does not adapt to the topology provided by the GNG and, thus, the recognition



**Figure 2.8:** Charts representing the mean results for experiment sets using the Spin Image descriptor.



**Figure 2.9:** Results for the experiments by detector, descriptor and filter.

rate is almost negligible.

## 2.7 Conclusions

In this chapter, we have presented comparative study focus on determine how to deal with noisy point clouds, for object recognition. The experimentation uses two different filtering methods(GNG and VG), to reduce the noise effect in the clouds. In the same way, three keypoint detectors (US, H3D, ISS) and three 3D descriptors (SHOT, SI, FPFH) were evaluated in order to identify the combination that achieves the better results.

Experimental results showed that the presence of noise in a 3D point cloud reduces the performance in a recognition process. Therefore, it is necessary to use a method to reduce the noise and, consequently, to increase the recognition rate. Our results show that the use of the GNG method improves recognition rates, obtaining better results than those for Voxel Grid and for the raw clouds. GNG reduces the noise without losing significant information, and enables good recognition results. In addition, we identify that Uniform Sampling is the keypoint detector that achieves the best rates of recognition together with the SHOT feature descriptor.

As future work, we propose to continue evaluating more 3D detection and description methods to find better combinations of performance and noise robustness. We also plan to include RGB information in the GNG in order to give support to keypoints and descriptors which use color information.



Universitat d'Alacant  
Universidad de Alicante

# Scene Classification based on Semantic Labeling

---

This chapter describes the process to build an image descriptor, based on semantic annotations, which had been produced by a deep learning external tool. Furthermore, it shows the evaluation of this descriptor in scene classification tasks, and we also compared the descriptor with well-known image descriptors using several classification models. This chapter is organized as follows. Firstly, Section 3.1, provides an introduction to the scene classification dilemma. Secondly, Section 3.2 presents related works focused on the solution of the scene classification problem. Next, in Section 3.3, the formulation of the scene classification problem using semantic labels is presented. Whereas, Section 3.4 describes the different descriptors used in this study and presents a detailed explanation of the Clarifai system and their performance. In Section 3.5, the experimental results are presented, and a discussion is carried out in Section 3.6. Finally, in Section 3.7 the main conclusions and lines of future work are outlined.

## 3.1 Introduction

The scene classification or indoor place categorization problem could be defined as the problem of classifying an image as belonging to a scene

category on the basis of a set of predefined labels [Maron and Ratan, 1998]. This problem is closely related to the semantic localization one, and it helps to identify the surroundings of an agent, such as a mobile robot, by means of scene categories such as corridor or kitchen. Scene classifiers are also helpful for specific robotic tasks [Martínez-Gómez et al., 2014] such as autonomous navigation, high-level planning, simultaneous location and mapping (SLAM), or human-robot interaction.

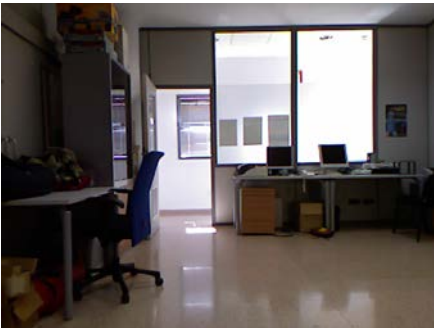
Scene classification is commonly addressed as a supervised classification process [Wu et al., 2009], where input data correspond to perceptions, and classes to semantic scene categories. Traditionally approaches are based on a two-stage building process: a) select the appropriate descriptors to be extracted from perceptions, and b) choose a classification model to be able to deal with the descriptors extracted.

Nowadays, available image descriptors are based on features such as texture or color that are present on an image, whereas, 3D descriptors usually focus on specific morphology in the environment, and also might present computational complexity in their calculations. On the other hand, these descriptors do not provide a human-readable description of the scene. With this in mind, in recent years deep learning techniques offer remarkably generalist classification tools, based on 2D image, whose output could briefly describe a location, in a semantic fashion. One of these tools is the Clarifai<sup>1</sup> web system, that taking advantage of large image collections, provides a system capable of accurately classifying an input image, using the previously learned semantic labels, as it is represented in the Figure 3.1. This contains the obtained values for the first 10 labels, produced by Clarifai for an image belonging to a research laboratory.

This chapter proposes a general framework for generating image descriptors from semantic labels. Specifically, we employ the annotation scheme provided by Clarifai, and then use these labels to build image descriptors. The descriptors obtained are evaluated as input for the scene classification problem. We have performed an exhaustive comparison with state-of-the-art global descriptors in the ViDRILO dataset [Martínez-Gomez et al., 2015].

---

<sup>1</sup><http://www.clarifai.com/api>

Image	Label	Probability
	Indoors	0.9936
	Seat	0.9892
	Contemporary	0.9787
	Chair	0.9779
	Furniture	0.9744
	Room	0.9634
	Interior design	0.9627
	Window	0.9505
	Table	0.9428
	Computer Technology	0.9417

**Figure 3.1:** Labels and probabilities obtained with the Clarifai API for a ViDRILO image.

The first goal of this chapter is then to determine whether Clarifai descriptors are competitive against other state-of-the-art image descriptors suitable for scene classification. It should be pointed out that Clarifai descriptors are generated from a general purpose labeling system. On the other hand, the rest of the image descriptors included in the experimentation have been specifically selected for their scene representation capabilities. This work is also aimed for discovering (and discussing) the novel capabilities offered by the use of general purpose annotations.

## 3.2 Scene Classification

Relying on the use of images as the main perception mechanism, the descriptor generation problem is tackled with computer vision techniques. In this process, the organization of the data extracted from the images plays an important role. This is clearly seen in two of the most widely-used approaches: the Bag-of-Words (BoW) [Csurka et al., 2004], and the spatial pyramid [Lazebnik et al., 2006]. These two approaches allow the generation of fixed-dimensionality descriptors, which are required for most state-of-the-art classification models, built from any type of local features.

There exist, however, novel approaches proposing the use of categorical



information instead of numeric image descriptors. For instance, [Li et al., 2010] proposes the use of an Object Filter Bank for scene recognition, where the bank is built upon image responses to object detectors that have been previously trained. [Lampert et al., 2014] presents an object recognizer based on attribute classification. This proposal relies on a high-level description that is phrased in terms of semantic attributes, such as the shape or the color of the object. The novel approach proposed in [Li et al., 2014] represents images by using the objects appearing in them. This high-level representation encodes both object appearances and spatial information.

### 3.3 Scene classification using semantic labels

The scene classification problem can be formulated as a classical statistical pattern recognition problem as follows. Let  $I$  be a perception (commonly an image),  $d(I)$  a function that generates a specific descriptor given  $I$ , and  $M$  a classification model that provides the class posterior probability  $P_M(c|d(I))$ , where  $c$  is a class label from a set of predefined scene categories  $\mathcal{C}$ . Then, this problem can be established as the problem of finding the optimal label  $\hat{c}$  according to:

$$\hat{c} = \arg \max_{c \in \mathcal{C}} P_M(c|d(I)) \quad (3.1)$$

In our case,  $I$  corresponds to an RGB image. The problem then involves two main stages: a) designing the descriptor generation process to obtain an appropriate representation of  $I$  ( $d(I)$ ), and b) selecting a classification model capable of discriminating among the set of predefined scene categories.

This chapter is focused on the first stage, namely descriptor generation, and we propose representing every image  $I_i$  as a sequence of semantic annotations obtained from an external labeling system, namely Clarifai. That is,  $d(I)$  is designed as a black box procedure where every image is translated into a set of  $N$  labels  $\mathcal{L} = \{l_1, \dots, l_N\}$  corresponding to the semantic annotations obtained from the Clarifai system. Labels can partially represent an input image  $I_i$ , and a set of probabilities  $\mathcal{P}_i = \{p_{i1} \dots p_{iN}\}$  is

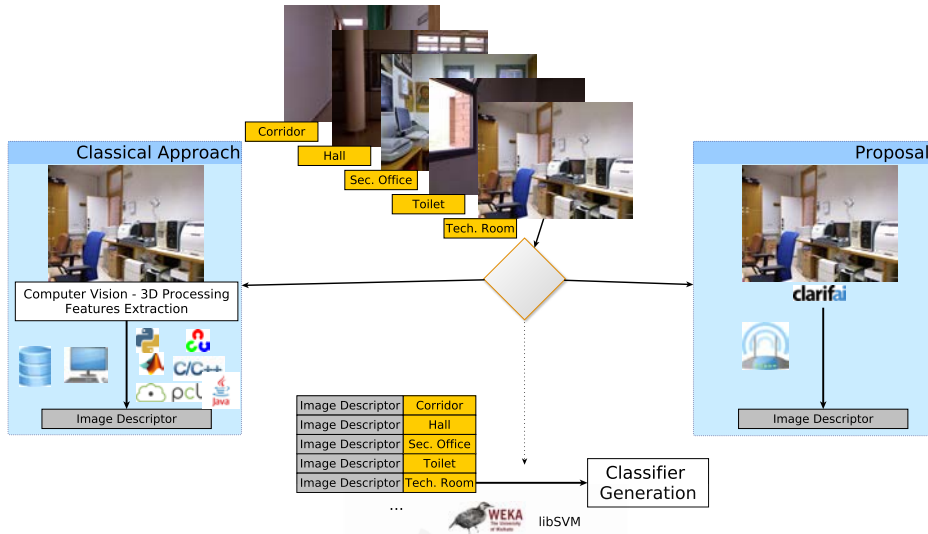
obtained in conjunction with  $\mathcal{L}$ . Each entry  $p_{i,j}$  represents the likelihood of describing the image  $I_i$  using the label  $l_j$ . Thanks to the use of a pre-defined set of semantic labels, we can obtain fixed-dimensionality image descriptors, as it will be explained in Section 3.4.

There exist some similarities between this approach and the classical Bag-of-Words [Csurka et al., 2004] one. In both representations, image descriptors consist of a set of terms describing the input perception. However, the main difference resides in the semantic component of our approach. That is, the dictionary of words (or codebook) in the BoW approach does not fully represent semantic concepts, as it is computed from a set of numeric local features in an unsupervised way, usually through a  $k$ -means clustering algorithm. In our case, the labels have a full semantic meaning.

### 3.4 Descriptor generation and description

This chapter proposes the use of general purpose Clarifai labels as input for a scene classifier. To evaluate this proposal, we carry out a comparison with classical approaches in which problem-oriented descriptors are directly computed from images. These two alternatives are shown in Figure 3.2. The classical approach extracts visual descriptors from the input images aiming for increasing their discriminating capabilities among scene categories. This stage should be carefully designed to select the appropriate features. In this design, we should also take into account aspects such as efficiency, the programming language or the requirements of external library dependencies.

On the other hand, the generation of descriptors from Clarifai labels is performed by delegating this step to an external system whose internal details do not need to be known. Moreover, no information about the problem to be faced, scene classification in this case, is provided to the annotation system. This would allow for the integration of any semantic annotation, but the results could be affected by the representation capabilities of the labels generated. As it is shown in Figure 3.2, the descriptors obtained are used as input for further classification tasks independently of the way in which they are generated.



**Figure 3.2:** Methodology in overall pipeline.

In order to validate this approach on the ViDRILO (see Section 1.4.3) dataset, we compare it against the baseline results obtained with the state-of-the-art descriptors presented in [Martínez-Gomez et al., 2015]. The feature extraction techniques, as well as the use of the Clarifai system, are detailed below.

### 3.4.1 Descriptor generation from visual and depth features

Three baseline descriptors are proposed and released in conjunction with ViDRILO. These baseline descriptors are: Pyramid Histogram of Oriented Gradients [Bosch et al., 2007] (PHOG), GIST [Oliva and Torralba, 2001], and Ensemble of Shape Functions [Wohlkinger and Vincze, 2011] (ESF). Both PHOG and GIST are computed from RGB images, while ESF relies on the use of depth information.

#### 3.4.1.1 Pyramid Histogram of Oriented Gradients (PHOG)

PHOG descriptors are histogram-based global features that combine structural and statistical approaches. This descriptor takes advantage of

the spatial pyramid approach [Lazebnik et al., 2006] and the Histogram of Gradients Orientation [Dalal and Triggs, 2005].

The generation of PHOG descriptors is shown in Figure 3.3, and it depends on two parameters: the number of bins of each histogram  $B$ , and the number of pyramid levels  $V$ . At each pyramid level  $v_i$  in  $[0 \dots V]$ , this process produces  $4^{v_i}$  histograms with  $B$  bins. According to the baseline ViDRILO experimentation, we choose  $B = 30$  and  $V = 2$ , which results in a descriptor size of 630  $((4^0 + 4^1 + 4^2) \times 30)$ .



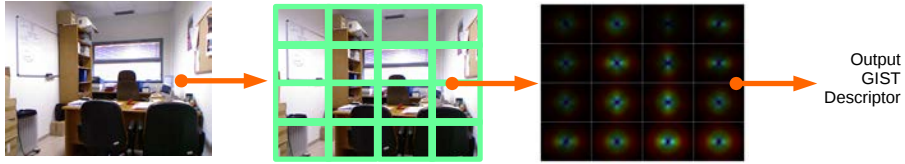
**Figure 3.3:** Generation of the PHOG Descriptor.

### 3.4.1.2 GIST

The GIST descriptor is intended to model the shape of a scene by using a holistic representation of the spatial envelope. In the GIST generation process,  $S \times O$  transformations are performed, with scales( $S$ ) and orientations( $O$ ), over  $N \times N$  patches in the image, as is shown in Figure 3.4. These transformations make it possible to represent each patch by a low-dimensional ( $S \times O$ ) vector, which encodes the distribution of  $O$  (orientations) and  $S$  (scales) in the image, together with a coarse description of the spatial layout. Using the standard implementation, the dimensionality of the GIST descriptors is 512 ( $N = 4, O = 8, S = 4$ ).

### 3.4.1.3 Ensemble of Shape Functions (ESF)

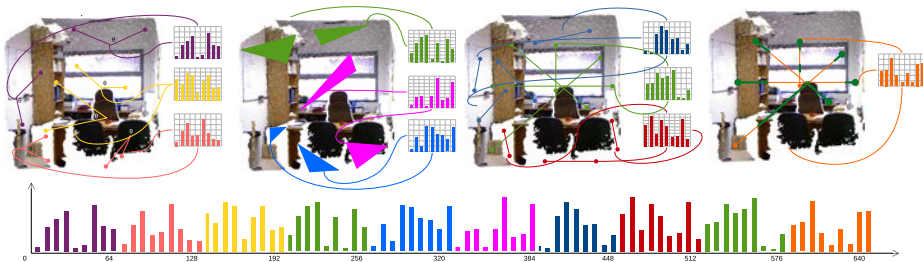
The Ensemble of Shape Functions (ESF) is a 3D point cloud global descriptor that consists of a combination of three different shape functions that result in ten concatenated 64-bin histograms.



**Figure 3.4:** Generation of the GIST Descriptor.

It is based on three shape functions describing distance (the distance between two randomly selected points), angle (the angle enclosed by two lines created from three randomly selected points) and area distributions (the area of the triangle formed by three randomly selected points). It also classifies each of the values into three classes based on where the connecting lines between points reside: *on* the object surface, *off* the surface and *mixed* (partly *on* and *off*). So the sub-histograms used to form the ESF are: 3 (*on*, *off*, *mixed*) for angle, 3 (*on*, *off*, *mixed*) for area, 3 (*on*, *off*, *mixed*) for distance, and a final one for the ratio of line distances between *off* and *on* parts of each considered line.

On the contrary to other 3D descriptors, ESF does not require normal information, which makes it robust to noise and partial occlusions. Each histogram contains 64 bins, and the final dimension of the descriptor is 640. Figure 3.5 shows the three histograms (*on*, *off* and *mixed*) obtained for the angle, area, and distance function, as well as the additional histogram, and the structure of the generated ESF descriptor.



**Figure 3.5:** Generation of the ESF Descriptor.

### 3.4.2 Descriptor generation from the Clarifai system

Clarifai [Clarifai, 2015] (see Section 1.6.1 for additional details) is a research project aimed at developing a high-level image (and video) processing system by means of Convolutional Neural Networks (CNNs) (see Section 1.5.1).

We propose the use of the Clarifai labeling system as a black box procedure through its application programming interface (API). From this API, we can automatically tag perspective images based on their content. It also provides the presence probability of each label in the image.

There exists a commercial version of the Clarifai API with non-restricted access to advanced features of the system. However, we use a free trial version that presents some limitations, such as the number of requests per hour and month. This version tags each input image with the 20 most feasible semantic labels, as well as their associated probabilities. This information is used to generate the image descriptors in this proposal.

The dimensionality-fixed descriptor generation process is carried out as follows. We firstly discover the total number of labels representing our dataset by:



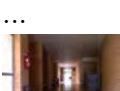

- Submitting all the images from our dataset to Clarifai.
- Identifying the unique values in the output label list.
- Encoding each image  $I_i$  by using a sparse representation whose dimension corresponds to the length of the entire list  $N = |\mathcal{L}|$ .

The descriptor consists of a set of entries  $\mathcal{W}_i = \{w_{i,1}, \dots, w_{i,N}\}$ . Each entry  $w_{i,j}$  contains the probability of representing  $I_i$  with label  $l_j$  only when the Clarifai response to this image request includes information for  $l_j$ . Otherwise, the entry  $w_{i,j}$  encodes the lack of information about  $l_j$  by using a zero value.

Using ViDRILO (1.4.3) as the dataset we obtain  $N = 793$  different labels. Therefore, each Clarifai descriptor has a dimensionality of 793, even when it contains information about only 20 different labels. An example of this process (using 4 instead of 20 labels per service request, for clarity)

is shown in Table 3.1. The semantic labels used in the Clarifai system represent meaningful concepts and the majority of them ( $\approx 80\%$ ) are nouns. Some exemplar labels are: animal, flower, plant, sport, structure, vehicle or person. Figure 3.6 presents a tag cloud for the 793 labels obtained for the whole ViDRILO dataset using Clarifai, the size of the word (semantic label) in the image represents the global probability ( $\bar{w}_{i,j}$ ) of the label in the entire dataset. Here, the outstanding labels such as: ‘room’, ‘building’, ‘hallway’, and ‘bathroom’ among others, give a brief description of the composition of the dataset.

**Table 3.1:** Clarifai descriptor generation from sparse semantic labeling

	$l_1$	$l_2$	$l_3$	$l_4$	$l_5$	$l_6$	$l_7$	...	$l_{793}$
	0.97	0.95	0.93	0.91	0	0	0	...	0
	0	0.94	0	0.92	0.93	0.94	0	...	0
	0.91	0	0	0	0.94	0.97	0.93	...	0
...	...	...	...	...	...	...	...	...	...
	0	0	0	0	0	0	0	...	0.91

## 3.5 Experimental framework

All the experiments included in this chapter have been carried out using all the sequences in the ViDRILO dataset (1.4.3). In the experimentation, several combinations of descriptors and classification models are evaluated in different scenarios. The main goal of the experiments is to determine the discrimination capabilities of Clarifai descriptors. Each scenario includes a training and a test sequence used to generate and evaluate the scene classifier, respectively. This evaluation computes the accuracy as the percentage of test images correctly classified with their ground truth category. Based on the five ViDRILO sequences, we are faced with 25 different scenarios (from Sequence 1 vs Sequence 1, to Sequence 5 vs Sequence





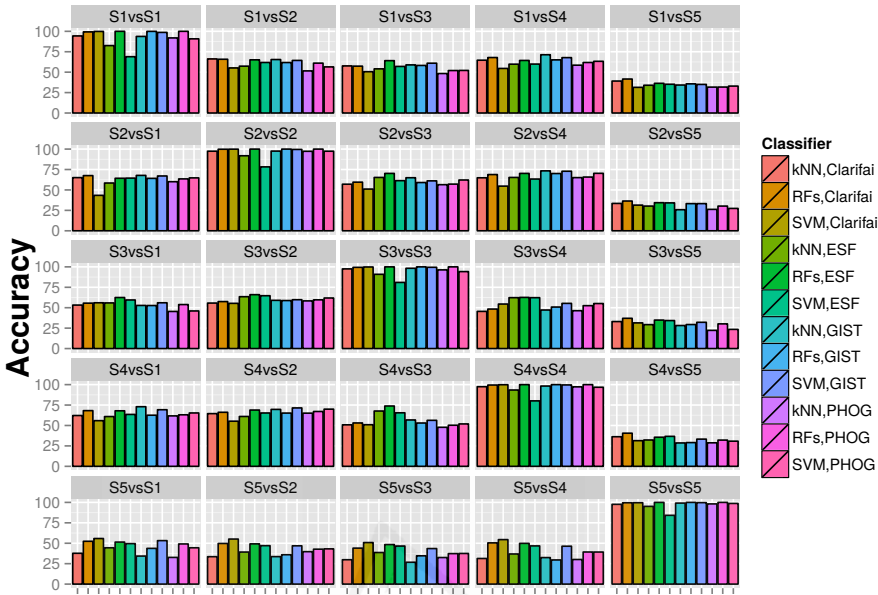
vice in a preliminary stage. However, the running time for Clarifai labeling greatly depends on the Internet connection and the overload of the system. This resulted in large time variations (in the range  $[0.15 - 1.25]$  seconds per image), and is due to the fact that there was no offline alternative to the online Clarifai labeling system.

### 3.5.1 Evaluation of Clarifai as visual descriptor

In the first experiment, we generated a Clarifai descriptor from all the images in the ViDRILO dataset. This process was carried out by following the methodology proposed in Section 3.4. Then, we integrated the Clarifai descriptors in the experimentation stage proposed by the dataset authors. This experimentation proposed the evaluation of the five sequences available in the dataset, using them firstly to train a classifier and then, using another sequence, to test the classifier. We followed this process using the combination of baseline classifiers and descriptors. The integration of Clarifai descriptors in the experimentation was performed by using them as input for the classifiers, as it was done for PHOG, GIST and ESF.

The results obtained are shown in Figure 3.7, where each chart title denotes the sequences used for the experiment. That is, *SAvsSB* means that sequence *A* has been used to train the model, whereas sequence *B* has been used to test it. The large amount of data makes it impossible to draw conclusions without a posterior analysis. Therefore, we post-processed the data to obtain the mean accuracy over the training/test combinations. Figure 3.8 graphically presents these results, grouping them according to the sequence combination. Figure 3.8 top shows the mean accuracy obtained when using the same sequence for training and test. Whereas, Figure 3.8 bottom shows the mean accuracy for every combination of different training and test sequences.

In order to carry out a fair comparison, we performed a Friedman test [Friedman, 1940] with a 0.05 confidence level. The null hypothesis established all the descriptor/classifier combinations as equivalent when faced with the scene classification problem. This hypothesis was rejected, which encouraged us to implement a post-hoc statistical analysis as described in [Demšar, 2006] to find out which of these combinations outper-

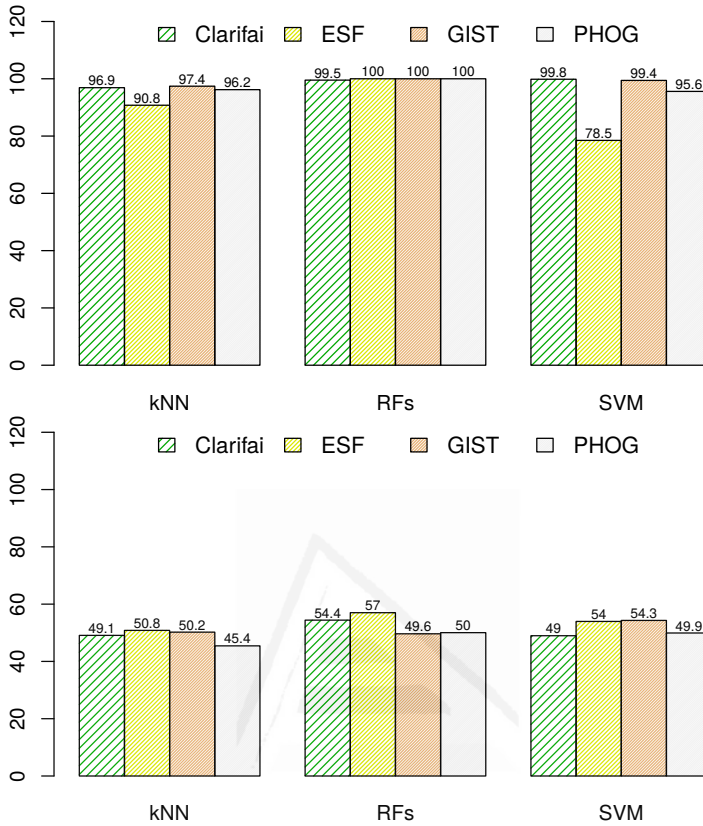


**Figure 3.7:** Accuracy obtained for all the classifiers, descriptors and training/test combinations.

formed the rest. The comparison was made against the best combination, namely ESF with RFs, and obtained the values that are shown in Table 3.2. In view of these results, we can state that by using Clarifai as descriptor, we can generate scene classifiers (using RFs as classifier) as competitive as those generated with two well-known descriptors such as ESF and GIST. It has also been shown how Clarifai enables results that are significantly better than those obtained with PHOG.

### 3.5.2 Coping with domain adaptation

If we review the specifications of the sequences included in ViDRILO, we find that Sequence 5 is the only one that has been acquired in a different building. Table 3.3 shows the difference among images of the Sequence 5 against the other sequences. Evaluating a scene classifier in an environment not seen previously makes the problem even more challenging. This has been revealed in Figure 3.7, where the poorest results are obtained whenever sequence 5 is used for training and any of the other sequences



**Figure 3.8:** Accuracy averaged over the training/test combinations, using the same sequence (top) and different sequences (bottom).

is used for testing, or vice versa (i.e. whenever the training sequence and testing sequence come from different buildings). These scenarios (where Sequence 5 appears) indicate the generalization capabilities of a scene classification system. That is, the knowledge acquired during training should be generalist enough to be applied to different environments. Therefore, we decided to perform an additional comparison between all the descriptors and classifiers in the 8 scenarios in which Sequence 5 is used for training or test. The results obtained are presented in Figure 3.9, where we can observe some interesting features. Firstly, all the methods ranked first by the scenario use Clarifai as descriptor. Moreover, these methods use a SVM classifier when using Sequence 5 for training (Figure 3.9 bottom), and a
















































**Table 3.2:** Post-hoc analysis comparison for all descriptor/classifier combinations against the best combination (ESF/RFs)

Descriptor,Classifier	p-value	Rejected	Rank	Win	Tie	Loss
RFs,ESF	-	No	2.70	-	-	-
SVM,GIST	2.2433e-01	No	4.22	19	0	6
RFs,Clarifai	2.2433e-01	No	4.32	14	0	11
SVM,ESF	1.5541e-03	Yes	6.24	23	0	2
RFs,PHOG	7.6841e-04	Yes	6.54	20	5	0
RFs,GIST	7.6841e-04	Yes	6.56	20	4	1
kNN,GIST	6.7068e-04	Yes	6.64	19	0	6
SVM,Clarifai	4.5236e-05	Yes	7.30	21	0	4
SVM,PHOG	2.4383e-05	Yes	7.46	22	0	3
kNN,Clarifai	9.3896e-06	Yes	7.68	20	0	5
kNN,ESF	2.7732e-06	Yes	7.94	25	0	0
kNN,PHOG	4.7705e-13	Yes	10.40	25	0	0

Random Forest classifier when this sequence is used for test (Figure 3.9 top). This figure shares the same notation as the one used in Fig 3.7. The difference between the classification models can be explained by the number of images included in the ViDRILO sequences (see Table 1.1 for details). That is, SVMs require more training instances than Random Forests to achieve proper discrimination capabilities. Consequently, SVMs perform better when trained from Sequence 5, which has 8,412 images in contrast to the rest of the sequences (3,510 images on average).

For these scenarios, we also carried out a Friedman test (0.05 confidence level) and a post-hoc statistical analysis. The null hypothesis that all descriptor/classifier combinations are equivalent was rejected. The post-hoc analysis was carried out against the best combination, namely Clarifai/RFs, and obtained the raking distribution shown in Figure 3.10. This comparison showed that just ESF (in conjunction with RFs and SVM) and the combinations of GIST and Clarifai with SVMs are not significantly different from Clarifai/RFs. Figure 3.10 illustrates the average rank position achieved with each combination of descriptor and classifier in the eight scenarios involving Sequence 5. It can be observed how the best combination always ranked between the first and fourth positions. This figure

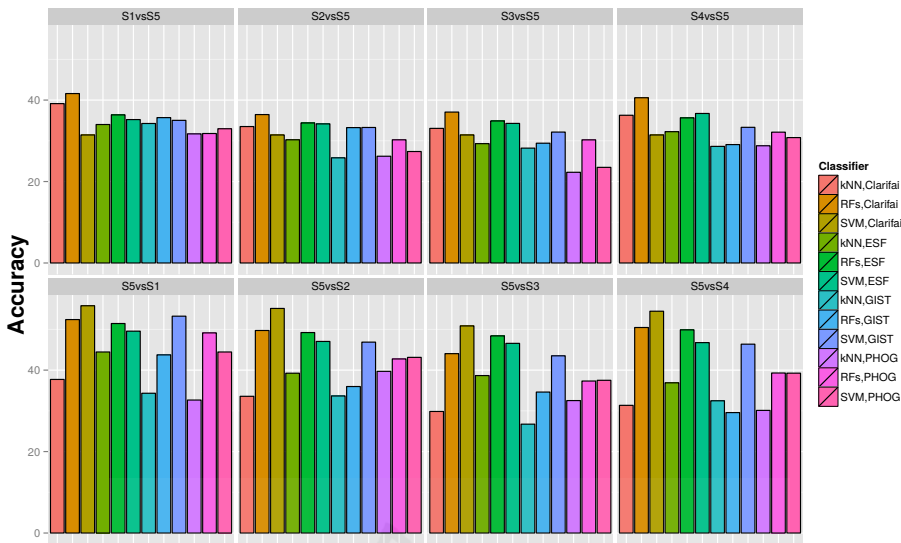
**Table 3.3:** Comparison of the images in the categories of the sequences in ViDRILO

Category	Sequence				
	1	2	3	4	5
Corridor					
Elevator Area					
Hall			N/A		
Professor Office					
Secretary Office					
Student Office					
Toilet					
Technical Room					
Video Conference Room			N/A		
Warehouse			N/A		

also helped us to discover the low discriminating capabilities of the kNN classifier.

### 3.5.3 Label Subset Selection

The experiments for this chapter included an additional step before the classification. This step consists of the application of a subset variable selection process in the Clarifai descriptors to reduce the number of attributes used to describe the image. In order to accomplish this reduction, the labels of the whole ViDRILO are averaged and decreasingly ordered. Using this order list of labels, new descriptors are produced selecting a TOP- $n$  number of labels, where  $n$  is the dimension of the new descriptor of the image.

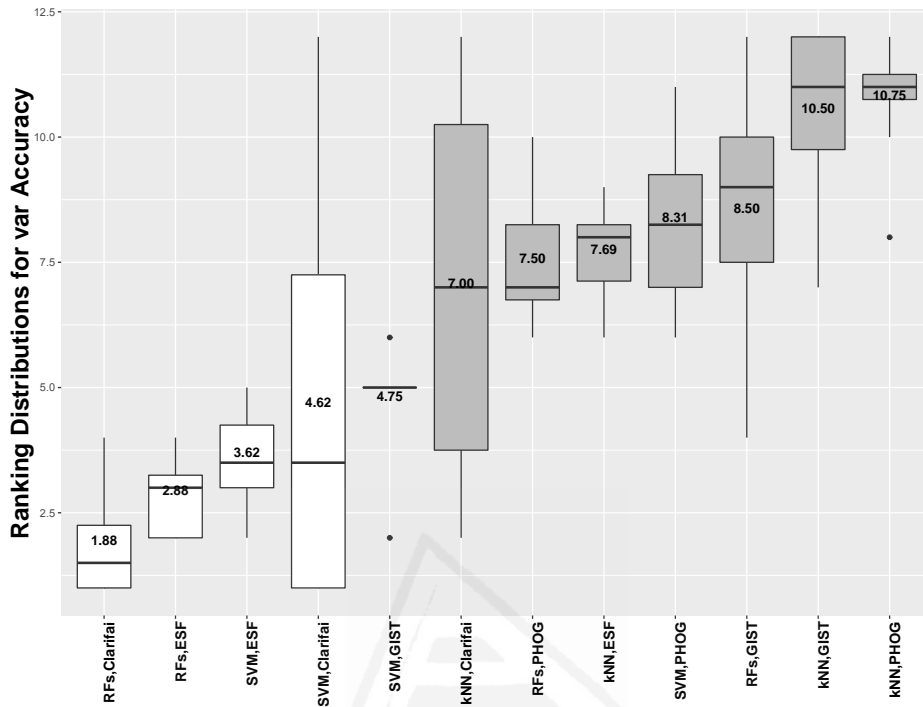


**Figure 3.9:** Accuracy obtained for all the classifier and descriptor combinations in those scenarios involving Sequence 5 as test (top) or training (bottom).

The descriptor still gets good results with this process, obtaining even better results using a reduced amount of labels to train the classifiers. Figure 3.11 shows the result of selecting 10, 20, 50, 75, 100, 150, 200 labels, and then training the SVM classifier. It shows that the use of a few labels reduces the accuracy results, but when the number increases the accuracy outperforms the one obtained using the 793 labels of the descriptor. Hence the use of 75 labels gets higher accuracy than the use of all the labels.

### 3.6 Discussion

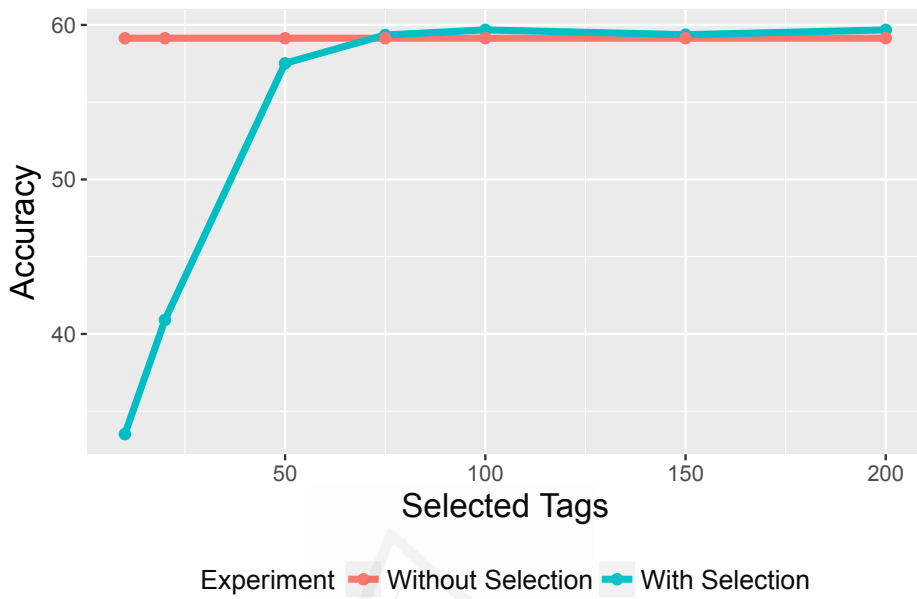
Semantic descriptors have been shown as an appropriate representation for the scene classification problem, with no significant differences with respect to global features such as GIST or ESF. However, two points can be highlighted if we review the semantic labels generated from ViDRILo images using, in this case, the Clarifai system. These labels are ranked by their distribution and graphically presented in Figure 3.12. We see that a small set of labels dominates the image descriptors. The frequency



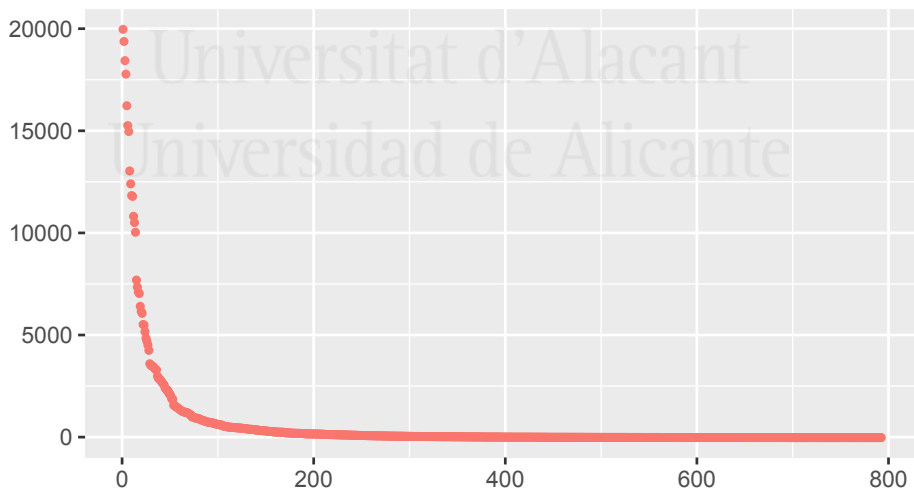
**Figure 3.10:** Average ranking for all classifier and descriptor combinations.

distribution shows that 50% of the annotations correspond to just 0.026% of the labels (21/793). This fact makes these labels play a very important role in the generation of the descriptor, to the detriment of the rest of the labels provided by the Clarifai annotation system.

A second point to be taken into account is the discrimination capabilities of this set of labels. To do so, we firstly selected only the 10 most frequent labels. From these labels, and taking advantage of another semantic annotations capability, we obtained some representative images of these concepts. Both, the labels and the representative images are shown in Figure 3.13, and we can observe how these labels are too generalist. In particular, these labels may be helpful for other computer vision tasks, such as determining whether an image represents an indoor or outdoor environment. However, these labels are not discriminating when faced with the scene classification problem. That is, the presence of the label “floor”



**Figure 3.11:** Accuracy of the Clarifai Descriptors using a Subset Variable Selection Process



**Figure 3.12:** Frequency obtained by the labels in ViDRILO.

does not help to resolve the type of scene where the image was acquired.





**Figure 3.13:** Exemplar visual images for the tags in the Clarifai descriptor.

In summary, the proposed semantic-based descriptor provides a very high degree of balance between simplicity and performance when compared against well-known complex image descriptors in the context of scene classification problems.

Also, the lexical nature of this descriptor, namely a set of labels describing the scene, allows for its direct human understanding. This interpretation of the descriptor can be used to integrate expert knowledge in the scene classification pipeline, such as expert-driven feature/label selection techniques, or high-order linguistic feature combinations using natural language processing techniques, among others.

### 3.7 Conclusions and future work

In this chapter, we have proposed and evaluated the use of semantic labels as a valid descriptor for the scene classification problem. These descriptors are generated from the semantic annotations obtained through an external API, in this case Clarifai. The trial version of the Clarifai API obtains the 20 most feasible labels representing the visual input image. Thanks to this approach, researchers can focus on selecting the appropriate classification model as the image descriptors are automatically generated.

In view of the results obtained, we can conclude that semantic labels descriptors are as competitive as state-of-the-art ones, which are computed with computer vision (GIST, PHOG), or 3D processing techniques (ESF). Moreover, proposed descriptor is shown to be the most outstanding descriptor when generalization capabilities are required. This situation has been evaluated by training scene classifiers from sequences acquired in a building, and then testing these classifiers on sequences acquired in a different building.

We have also reviewed the distribution, as well as the description, of the semantic labels obtained with Clarifai from ViDRILO images. In this review, it has been shown that we are tackling a specific problem (scene classification) using generalist information. That is, Clarifai annotations are not focused on describing scenes, but general concepts. Despite this fact, a competitive image descriptor has been proposed, developed and evaluated.

As future work, we plan to select a set of relevant semantic labels in a preliminary step, and then perform problem-oriented labeling by querying Clarifai about the probabilities of these labels in the image. We also have in mind the use of classifiers capable of working with missing values.

Furthermore, so far we are using only the Clarifai responses, namely the final layer in the CNN architecture. This is because Clarifai does not provide internal CNN feature values. We intend to change our system to use an open framework, such as Caffe, with which we are able to use internal layers values. Thus we plan to use the fully connected layers for classification purposes.



Universitat d'Alacant  
Universidad de Alicante

# LexToMap: Lexical-based Topological Mapping

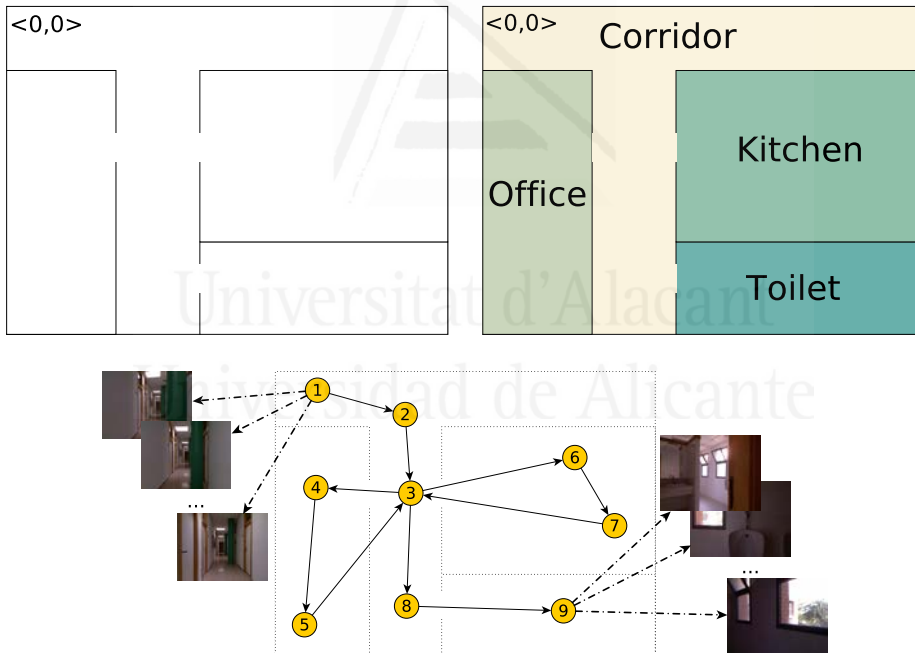
---

This chapter presents a generalist framework to build descriptive topological maps using lexical labels, generated with CNNs, as a representation of an image. This chapter is organized as follows. Firstly, Section 4.1 introduces the mapping problem, their aspects and the relevances for the robotics fields. Next, Section 4.2 reviews some related works and state-of-the-art solutions to the topological mapping problem. The process for extracting annotations and computing the similarity between images based on lexical labels is presented in Section 4.3. Then, Section 4.4 describes the procedure for generating topological maps from lexical labels. Experimental results and the descriptive capabilities of the LexToMap proposal are presented in Section 4.5. Finally, the main conclusions of this chapter as well as some future research directions are outlined in Section 4.6.

## 4.1 Introduction

Building an appropriate representation of the environment in which an autonomous robot operates is still a widely addressed problem in the robotics research community. This problem is usually known as map building or mapping since maps are considered the most common and appro-

appropriate environment representation [Thrun et al., 2002]. A map is useful for robot localization, navigation [Booi et al., 2007] and path-planning tasks [Bhattacharya and Gavrilova, 2008], but also for a better understanding of the robot's surroundings [Pronobis et al., 2010]. That is, a map may not be limited to metric (e.g. specific poses of objects/obstacles) and topological information (e.g. paths from one place to others), but it can also integrate semantic information (e.g. symbolic representations of objects, expected behaviors for specific locations, or even situated dialogues, to name a few) corresponding to the objects, agents, and places represented on it. Three different type of maps are graphically presented in Figure 4.1, where can be observed the bridge between metric and semantic representations.



**Figure 4.1:** Metric (top-left), metric-semantic (top-right), and topological exemplar maps (bottom)

Topological mapping consists in generating a graph-based representation of the environment, where nodes represent locations and arcs transitions between adjacent locations [Fraundorfer et al., 2007]. When using

images as input data the topological map construction process requires several image-to-image or image-to-nodes (set of images) comparisons in order to incrementally build the topological map.

This problem has been widely studied in robotics, and most of the state-of-the-art approaches rely on the use of computer vision techniques to estimate the similarity between robot perceptions, which are usually in the form of images [Valgren et al., 2006, Angeli et al., 2009]. This standard approach, however, presents an important drawback: the poor interpretability of the generated maps. Furthermore, two images can be visually different while representing a similar location, due to changes in the viewpoint or structural modifications.

To cope with these two drawbacks, in this chapter we propose the use of image annotations as input for topological map generation instead of usual visual features extracted from the image. By image annotations we refer to the set of tags or lexical labels used to characterize an input image. While the annotation process has been traditionally an expensive or even unapproachable task, the recent availability of deep learning models allows for efficient real-time annotations of any input image. These models are trained using huge datasets, such as ImageNet [Deng et al., 2009, Russakovsky et al., 2015] or Places [Zhou et al., 2014], where images are annotated using a large and heterogeneous set of lexical labels.

The advantages of using lexical labels to describe/represent an image (in our case obtained from deep learning classification models) are twofold:

- First, the similarity between images can be computed without the use of any computer vision technique. That avoids selecting the optimal set of image features to be extracted (e.g. SIFT [Lowe, 1999], SURF [Bay et al., 2006], HoG [Dalal and Triggs, 2005], ...), to make use of dimensionality reduction techniques, as well as carrying out further parameter tuning processes, which typically rely on proposals that are too specific and environment dependent.
- Second, the locations or nodes of the generated topological map can be described by means of the lexical labels associated to their contained images.

This novel map representation allows automatic objective-driven navigation, since a robot can understand a sentence such as “*bring me a cup of coffee*” without the need of making any explicit reference to the location where coffee cups are expected to be (typically in the kitchen) or where the beneficiary of the action is currently located.

The main contribution of the work presented in this chapter is the generalist framework for generating descriptive topological maps. The proposal has been evaluated on the KTH-IDOL 2 dataset, which consists of sequences of images acquired under three different lighting conditions: sunny, cloudy, and night. Moreover, the descriptive capabilities of the maps have also been shown and discussed for future applications.

## 4.2 Topological Mapping

The similarity between images has been widely used for several robotic tasks such as object recognition [Cyr and Kimia, 2001], navigation [Tudhope and Taylor, 1997] and semantic localization [Martínez-Gómez et al., 2011]. Regarding topological mapping, large image collections [Fraundorfer et al., 2007] are the traditional main source of information. This fact increases the computational time when applying image matching approaches, and this encourages the search for alternative approaches, capable of coping with large sequences of images. The visual similarity between images has traditionally been computed from invariant local features [Goedemé et al., 2005, Valgren et al., 2006], and global image descriptors [Koseck and Li, 2004, Liu et al., 2009], mainly generated by following bag-of-words approaches [Filliat, 2007]. From these image representations, the spatial distribution of the map has been modeled using graph representations [Cummins and Newman, 2008], as well as hierarchical proposals [Kuipers et al., 2004]. More concretely, [Cummins and Newman, 2008] provides a way to detect loop closure, but the proposed system needs to learn the visual features in the environment. Our method differs from the former in twofold: first, we do not need to learn the environment and, second, our aim is not only to detect loop closure but also to build the map at the same time, which is not achieved by [Cummins and Newman, 2008].

OpenRatSlam [Ball et al., 2013] and ORBSlam [Mur-Artal et al., 2015] are well-known current SLAM solutions, which rely on the use of matching and bag-of-words approaches respectively, but their requirements (visual images should be provided in conjunction with the camera rotational and translational velocity) and limitations (poor descriptive capabilities of the generated maps) encourage the search for novel approaches related to topological mapping.

The emergence of deep learning in the robotic community has opened up new research opportunities in the last few years. In addition to model generation for solving open problems [Bo et al., 2013, Neverova et al., 2014], the release of pre-trained models allows for a direct application of the deep learning systems generated [Rangel et al., 2016a], described in Chapter 3. This is possible thanks to the existence of modular deep learning frameworks such as Caffe [Jia et al., 2014] (see Section 1.6.2 for additional details). The direct application of pre-trained models avoids the computational requirements for learning them: long learning/training time even using GPU processing, and massive data storage for training data. From the existing deep learning models, we should point out those generated from images categorized with generalist and heterogeneous lexical labels [Krizhevsky et al., 2012, Zhou et al., 2014]. The use of these models lets any computer vision system annotate input images with a set of lexical labels describing their content, as it has been recently shown in [Carneiro et al., 2015, Murthy et al., 2015, Rangel et al., 2016a].

### 4.3 Lexical-based Image Descriptors

In contrast to most of the topological mapping proposals, we describe or represent images by means of a set of predefined lexical labels. The rationale behind this representation is to describe the content of the image by means of a set of semantic concepts that can be automatically attributed to this image. For example, if we describe an image saying that the appearance in it of concepts such as fridge, table, chair, cabinet, cup, and pan, is much more likely than other different concepts in the predefined set, then we can say that the image represents a kitchen with a high



degree of confidence. The use of lexical labels may result into a loss of resolution suitable for increasing the perceptual aliasing problem [Chrisman, 1992]. Besides fine grain representations, by means of large sets of labels in our proposal, the perceptual aliasing problem is reduced by taking into account the temporal continuity of the sequence. This is expected to associate different locations to different nodes, even when both are translated into similar descriptors.

To implement the lexical annotation process we make use of existing deep learning annotation tools. Deep learning techniques, and more specifically Convolutional Neural Networks (CNNs) [Lee et al., 2009] (see Section 1.5.1 for details), allow the generation of discriminant models while discovering the proper image features in a totally unsupervised way, once the network architecture has been defined. This is possible nowadays thanks to the availability of huge image datasets annotated with large and miscellaneous set of lexical labels, which efficiently permit the training of these discriminative classification models. In this chapter, we focus on the application of existing CNN models. The definition and building of these CNN models is beyond the scope of this chapter, so we refer the reader to [Bengio, 2009] for a more detailed view of deep learning in general and, to [Jia et al., 2014] for a better understanding of these CNN models.

Once every image is represented by a set of lexical labels, we need to define a similarity measure between two image descriptors or between an image descriptor and a node descriptor. A node on the topological map is composed of a set of images representing that node/location.

The complete process of annotating images using CNNs and the similarity computation details are described below.

### 4.3.1 Image annotation using CNN

Let  $\mathcal{L} = \{l_1, \dots, l_{|\mathcal{L}|}\}$  be the set of  $|\mathcal{L}|$  predefined lexical labels,  $I$  an image, and  $N$  a node of the topological map formed of  $|N|$  images. The direct application of the existing pre-trained CNN models on an input image  $I$  generates a descriptor defined as:

$$d(I) = ([p_I(l_1), \dots, p_I(l_{|\mathcal{L}|})]) \quad (4.1)$$

where  $p_I(l_i)$  denotes the probability of describing the image  $I$  using the  $i$ -th label in  $\mathcal{L}$ . This obtains a representation similar to the Bag of Visual Words (BoVW [Sivic and Zisserman, 2003, Martínez-Gómez et al., 2016]) approach, which generates a descriptor vector  $d_{BoVW}(I) = [n_I(w_1), \dots, n_I(w_k)]$  of  $k$  visual words, where  $n(w_i)$  denotes the number of occurrences of word  $w_i$  in image  $I$ . Despite the fact that spatial relation between words is completely removed, we decide not to follow any of the proposed techniques, like the spatial pyramid [Lazebnik et al., 2006], to solve this drawback. In addition to avoid the higher processing requirements this technique demands, our technique relies in the assumption that the presence of lexical labels is much more important than their position to describe any input image.

We use a similar notation to represent the descriptor of a node  $N$  of the topological map, which is composed of a set of  $|N|$  images ( $N = \{I_1, \dots, I_{|N|}\}$ ). The descriptor of  $N$  is defined as the vector of the average label probability of its  $|N|$  images, and the corresponding vector of standard deviations. More formally:

$$d(N) = ([\bar{p}_N(l_1), \dots, \bar{p}_N(l_{|\mathcal{L}|})], [\sigma_N(l_1), \dots, \sigma_N(l_{|\mathcal{L}|})]) \quad (4.2)$$

where:

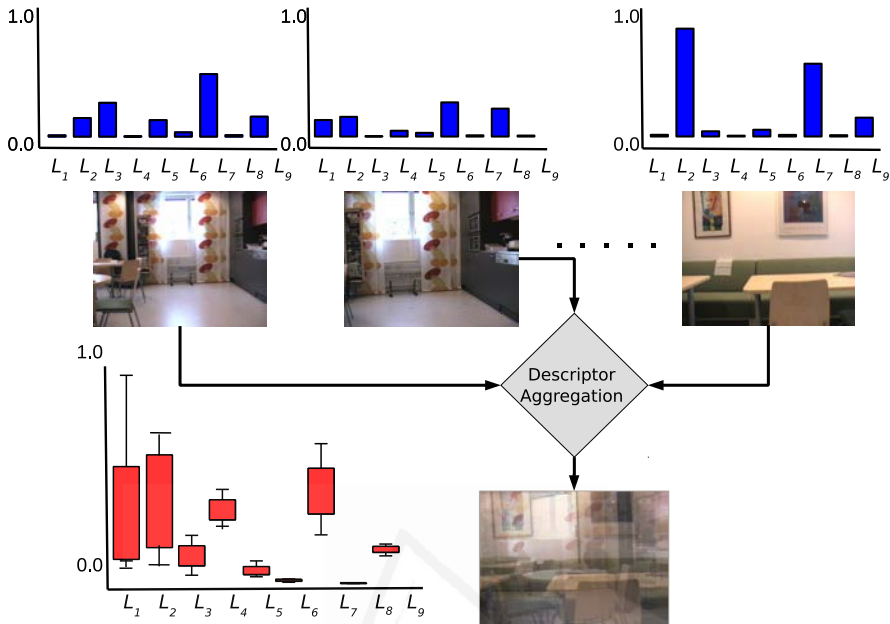
$$\bar{p}_N(l_i) = \frac{1}{|N|} \sum_{j=1}^{|N|} p_j(l_i) \quad (4.3)$$

and

$$\sigma_N^2(l_i) = \frac{1}{|N|} \sum_{j=1}^{|N|} (p_j(l_i) - \bar{p}_j(l_i))^2 \quad (4.4)$$

This average computation is actually the aggregation of all image descriptors that form the node. In Figure 4.2 a visual interpretation of this aggregation process is shown.

While the  $i$ -th average values encode the probability of describing the location using the lexical label  $i$  (e.g. the probability of describing the location as “table” is 75%), the standard deviation indicates whether this label is representative of the location. That is, large deviations denote lexical labels whose likelihood is not constant in the images from the same



**Figure 4.2:** An example of the aggregation process of  $n$  different images to define the descriptor of a node/location of only nine lexical labels

location. Therefore, we propose to integrate this information into the descriptor definition of the node in order to reduce the importance of lexical labels with large standard deviations, which are eventually considered not representative of the node/location.

### 4.3.2 Image/Node Descriptor Similarity Computation

On the one hand, the similarity between two images,  $I_a$  and  $I_b$ , whose representation has been obtained from a pre-trained CNN model, can be estimated using histogram intersection techniques [Lee et al., 2005]. In this case, we need to compare two descriptors,  $d(I_a)$  and  $d(I_b)$ , encoding the set of likelihoods describing images  $I_a$  and  $I_b$  using the set of predefined lexical labels. We can adopt well-known similarity measures such as  $p$ -norm based distances (i.e. Manhattan or Euclidean distance), the Bhattacharyya or the  $\chi^2$  distance, among others, to compare them.

On the other hand, the similarity between an image  $I$  and a node

$N$  is defined in this proposal as a weighted similarity using the standard deviation of the labels ( $\sigma_N(l_i)$ ) within the node. This is done to explicitly reduce the importance of labels presenting large variance in the node, which are considered non-representative ones, as well as to increase the relevance of those labels with low variance.

Based on the weighted euclidean distance formulation, the distance between a node/location  $N$  and an image  $I$  is computed according to:

$$D(N, I) = \frac{1}{\sum_{i=1}^{|\mathcal{L}|} (w_i)} \sum_{i=1}^{|\mathcal{L}|} (w_i \cdot (\bar{p}_N(l_i) - p_I(l_i))^2) \quad (4.5)$$

where  $w_i$  has been defined to be inversely proportional to the standard deviation and normalized in the range  $[0, 1]$ .

## 4.4 LexToMap: Lexical-based Topological Mapping

From the image descriptors obtained from pre-trained CNN models, and using the distance functions described above to estimate the distance between an image and a location, we define the lexical-based topological mapping using the pseudo-code in Algorithm 4.1 and graphically shown in the Figure 4.3.

In this process, we can find the starting situation where a new node (representing a location) is created from the first image. From there, we firstly try to add the images to the current node in order to take advantage of the temporal continuity of the sequence. If this is not possible, due to a big difference between the image and the current node (using threshold  $\tau_1$ ), we search in the node list for the most similar node. If this node exists, and it is similar enough to the image (using threshold  $\tau_2$ ), we mark it as the current one, we add the image to it, and create a transition (edge) from the former node to the current one, if it does not already exist. Otherwise, we create a new node on the map, which is established as the current one, and then the transition from the past node to the new one is created.

Each node or location consists of a set of image descriptors encoded

**Algorithm 4.1:** LexToMap: Lexical-based Topological Mapping

---

```

Data: CurrentNode = None
output: NodeList =  $\emptyset$ 
1 for each image  $I_j$  acquired from the robot do
2   if  $\text{length}(\text{NodeList}) == 0$  then
3     Create a new Node  $N_{new}$  from  $I_j$ 
4     Add  $N_{new}$  to NodeList
5     CurrentNode =  $N_{new}$ 
6   else
7     if  $D(\text{CurrentNode}, I_j) < \tau_1$  then
8       CurrentNode = CurrentNode  $\cup I_j$ 
9     else
10       $N_{sim} = \text{None}$ 
11       $Min_{dist} = \infty$ 
12      forall node  $N_z$  in NodeList do
13         $d_z = D(N_z, I_j)$ 
14        if  $d_z < Min_{dist}$   $\&\&$   $N_z \neq \text{CurrentNode}$  then
15           $N_{sim} = N_z$ 
16        end if
17      end forall
18    end if
19  end if
20  if  $N_{sim} \neq \text{None}$   $\&\&$   $D(N_{sim}, I_j) < \tau_2$  then
21    Create a transition from CurrentNode to  $N_{sim}$ 
22    CurrentNode =  $N_{sim}$ 
23    CurrentNode = CurrentNode  $\cup I_j$ 
24  else
25    Create a new Node  $N_{new}$  from  $I_j$ 
26    Add  $N_{new}$  to NodeList
27    Create a transition from CurrentNode to  $N_{new}$ 
28    CurrentNode =  $N_{new}$ 
29  end if
30 end for

```

---

as vectors representing lexical label probabilities. For evaluation and visualization purposes, we can also identify a node by its  $\langle x, y \rangle$  position in the environment by taking advantage of the ground truth of the dataset. The coordinates of a node are represented by the average values of  $x$  and  $y$  computed from the position coordinates of all the images included in the node.

The topological maps generated with our proposal would be trajectory dependent, as the first image acquired with the robot plays a very impor-

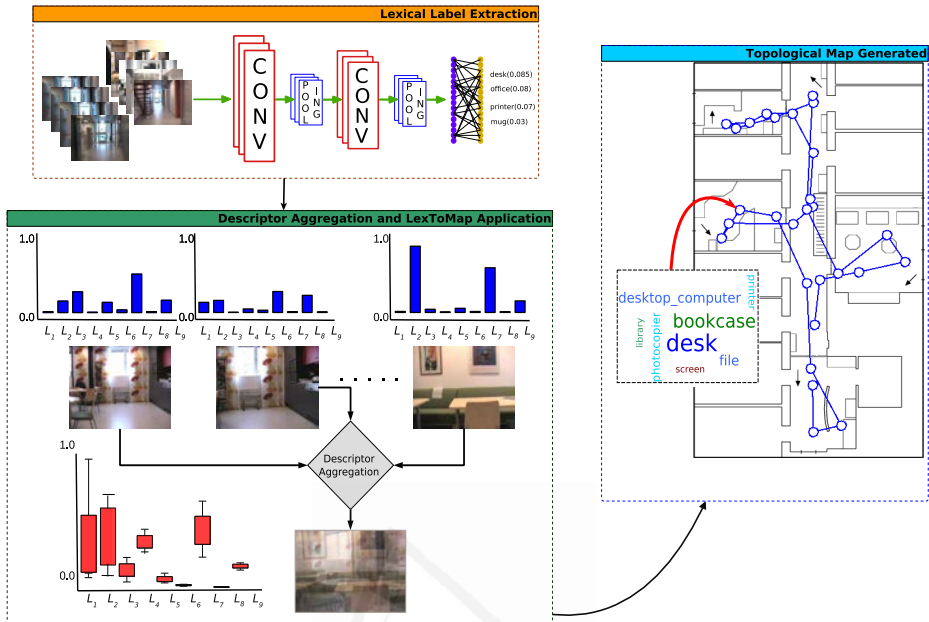


Figure 4.3: Scheme of the LexToMap proposal.

tant role in the process. The temporal continuity is also exploited to reduce the perceptual aliasing problem. Nevertheless, this dependency also allows the mapping procedure to generate maps in an online fashion. This avoids waiting for further acquisitions for making decisions about nodes and transition generation, which is undesired for any robotic system. Moreover, the online generation of topological maps permits the robot to return to intermediate previous locations. This situation is commonly faced due to battery problems, when the robot should come back as soon as possible to the charging area. Rescue robots may also cope with similar scenarios, where the riskiness of a discovered area encourage the robot to return to previous safe locations.

## 4.5 Experimental Results

The LexToMap topological map generation approach was evaluated under the three different lighting conditions proposed in the KTH-IDOL

2 dataset (Section 1.4.2). We followed the procedure explained in Algorithm 4.1, which starts from the descriptor generation procedure. This steps relies on the use of a CNN with a pre-trained model, and we had evaluated seven different model alternatives (see Table 1.2 in Section 1.6.2.1).

The procedure depends on two different thresholds,  $\tau_1$  and  $\tau_2$ , which determine the generation of new nodes and the transitions between them. All these steps will be detailed in the following subsections.

#### 4.5.1 Dataset

The experimentation carried out in this chapter have been developed using the KTH-IDOL 2 dataset (see Section 1.4.2) specifically the People-Bot sequences, with their three lighting conditions available in the dataset (cloudy, night and sunny). This selection gives us a total of twelve different sequences to test the LexToMap approach.

#### 4.5.2 Model Selection

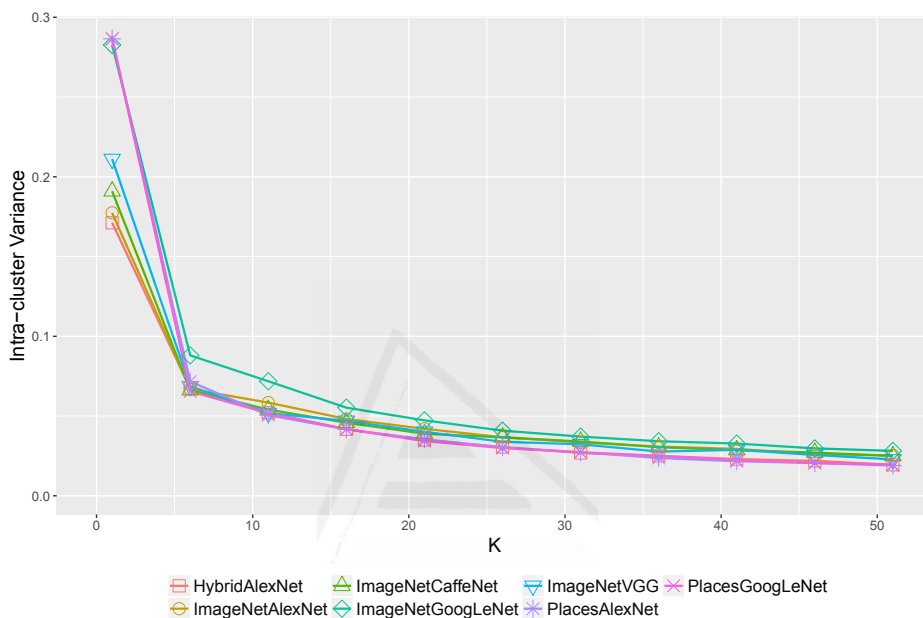
The experimentation of this chapter evaluates seven models among all the available in the Caffe Model Zoo<sup>1</sup>, owing to these models where trained with datasets relative to objects and scenes namely ImageNet(see Section 1.6.4.1), Places205 (see Section 1.6.4.2) and Hybrid (see Section 1.6.4.3). Therefore, their categories contain relevant lexical concepts for the images in the KTH-IDOL 2 dataset. The details of the selected models are presented in the Table 1.2 in the Section 1.6.2.1.

In this experimentation, we are interested in the categorization capabilities of the lexical labels generated through the CNN models. Therefore, we firstly propose an unsupervised learning procedure using the dataset sequences as input. This was carried out by using a  $k$ -means clustering algorithm considering different values of  $k$  in the range  $[1, 50]$ . Each cluster represents a location (topological node) computed using only image similarity information. That is, the temporal continuity of the sequence is not taken into account. Then, we evaluate the average spatial intra-cluster variance by using the dataset ground truth  $\langle x, y \rangle$  location of the

---

<sup>1</sup><https://github.com/BVLC/caffe/wiki/Model-Zoo>

input images. Figure 4.4 graphically presents the evolution of the intra-cluster variance when using different values of  $k$  in the  $k$ -means clustering algorithm. This figure introduces a bias-variance trade-off where larger  $k$  values result in less generalist but more accurate clusters.



**Figure 4.4:** Intra-cluster spatial evolution using different values of  $k$  for the 7 CNN models studied

Table 4.1 shows a subset of the results obtained using four representative values of  $k$ . In this table, each column presents the average spatial intra-cluster variance of the whole combination of lighting conditions and sequences for a specific value of  $k$ . Lower intra-cluster variances are desirable as they denote a more precise representation of the data. Indeed, low variances are obtained with clusters that consist of images acquired from nearby environment positions. From the complete set of results, we computed the average ranking for all values of  $k$  in the range  $[1,50]$ . That resulted in 50 different test scenarios where each model was ranked between the first and the seventh position, depending on its intra-cluster variance.

The ranking comparison summary is presented in Figure 4.5, where it can be observed how Hybrid-AlexNet clearly outperforms the rest of the



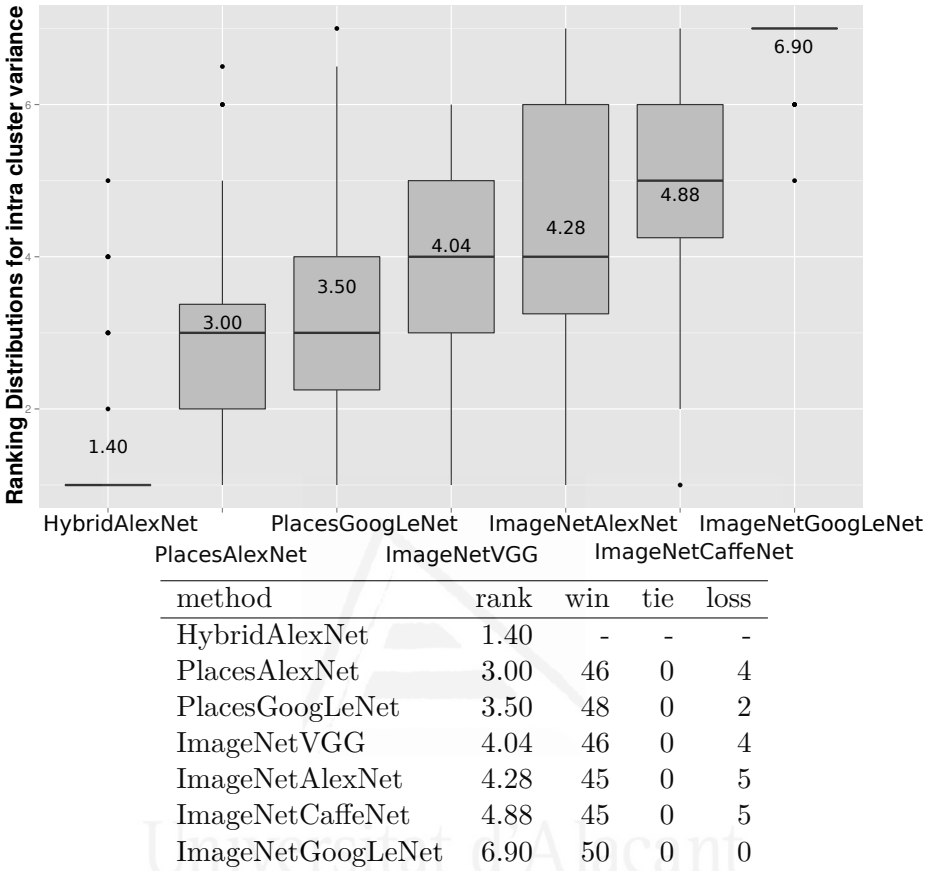
**Table 4.1:** Intra-cluster variance ( $\cdot 10^{-2}$ ) for 7 CNN models and four representative  $k$  values. Lowest values per column are in bold.

CNN Model	k=7	k=15	k=30	k=50
HybridAlexNet	<b>5.49</b>	<b>3.99</b>	<b>2.50</b>	<b>1.85</b>
ImageNetAlexNet	7.53	4.19	3.04	2.42
ImageNetCaffeNet	6.24	5.34	3.23	2.43
ImageNetGoogLeNet	7.35	6.02	3.97	2.90
ImageNetVGG	5.69	4.44	3.17	2.39
PlacesAlexNet	6.64	4.37	2.90	1.99
PlacesGoogLeNet	6.91	4.59	2.85	1.97

evaluated models. Therefore, we selected this model as optimal (among those used in this study) for topological mapping, and it was used for the rest of the experimentation. The proper behavior of the Hybrid dataset comes from the fact that it has been generated from a combination of both Places and ImageNet datasets, once the overlapping scene categories were removed [Zhou et al., 2014]. The ranking comparison also pointed out the appropriateness of using the Places dataset in contrast to ImageNet, which is explained due to the nature of the annotations, more suitable for discriminating between indoor locations. With regard to the network architecture, those with lower number of convolution layers presented the best behaviors.

### 4.5.3 Topological Map Generation

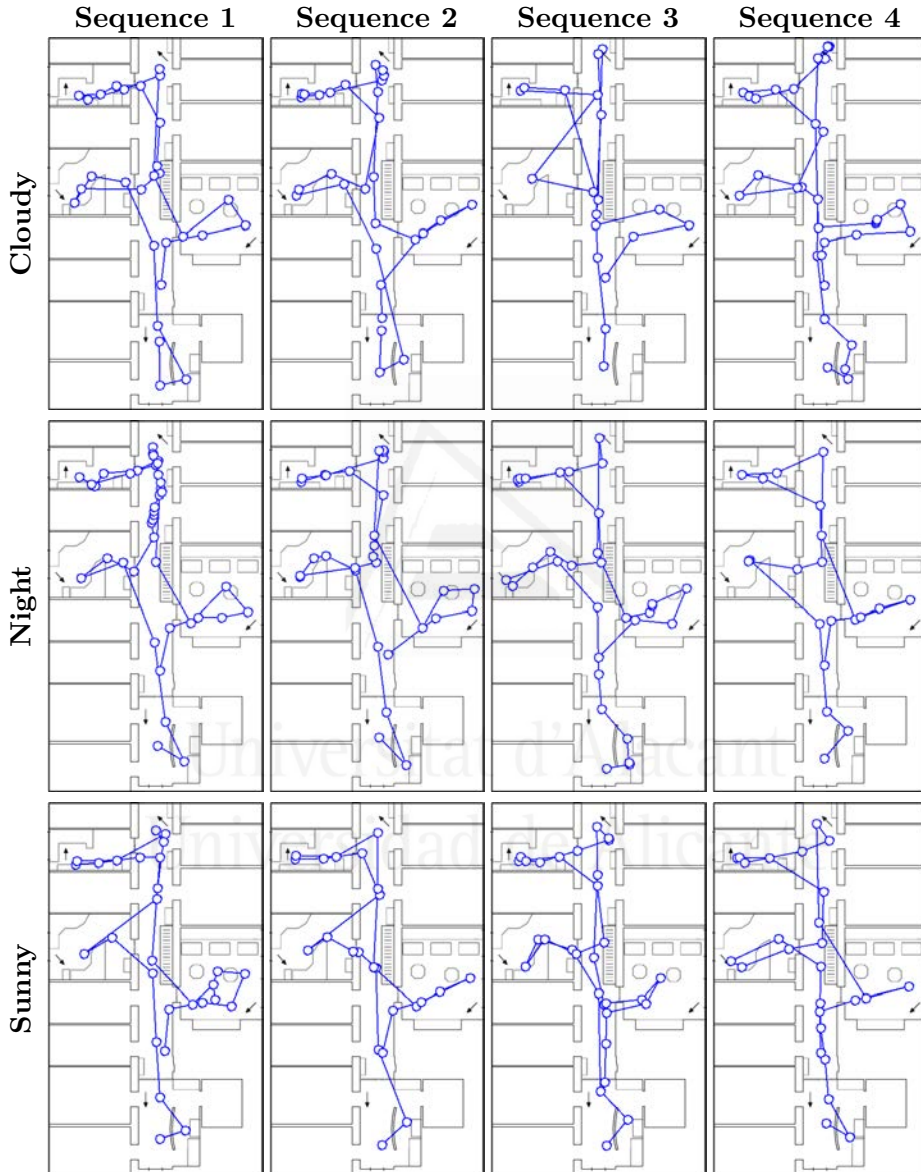
Once the pre-trained CNN model has been chosen, we should establish the values for the two thresholds used in the topological mapping:  $\tau_1$  and  $\tau_2$ . These threshold values can be selected to generate different types of maps based on the end requirements of the topological maps. For example, we can prevent the algorithm from generating a large set of nodes/locations by selecting large  $\tau_1$  values. And, large  $\tau_2$  values facilitate the generation of connections between existing nodes. This increases the average number of connections by node. The automatic selection of these two thresholds would require the availability of a quantitative metric to evaluate the goodness of any topological map. Unfortunately, we could not find any proven metric in the literature and its generation is not triv-



**Figure 4.5:** Ranking comparison of 7 CNN models. Win, tie and loss results (bottom) and average ranking visualization (top)

ial. In order to establish a trade-off between specificity and generality, we empirically selected  $15 \cdot 10^{-2}$  and  $15 \cdot 10^{-3}$  for  $\tau_1$  and  $\tau_2$  thresholds, respectively. Using the Hybrid-AlexNet CNN model, we generated a set of twelve topological maps for a more detailed evaluation and discussion from all the dataset sequences. All these maps were generated using the same internal parameters (pre-trained CNN model and thresholds).

The maps generated are shown in Figure 4.6 for three lighting conditions. It can be observed how valuable topological maps can be generated thanks to the use of the LexToMap proposal without the need for any other additional or complementary visual information. Although lighting variations within indoor environments are not so challenging as for outdoor



**Figure 4.6:** Topological maps generated for three different lighting conditions: cloudy (top), night (middle) and sunny (bottom)

ones, we opted for an indoor dataset incorporating some lighting changes (see Figure 1.4). The maps generated are not drastically affected by these changes thanks to the use of the lexical labels to compute the similarities between images, which are proposed instead of standard visual features.

In Figure 4.7, we can observe two different types of transitions, which correspond to the generation of the map from sequence 3 acquired with cloudy lighting conditions. Concretely, we illustrate the timestamps when the robot backs to the corridor after visiting the one-person office. During previous tours along the corridor, the algorithm created different nodes and transitions between them. Before leaving the one-person office, the mapping algorithm has Node 10 as current node. When the robot acquires an image different from previous ones (Figure 4.7 bottom right), Node 4 is discovered as an aggregation of images similar to the last robot perception. This is translated into a new transition between nodes 10 and 4. After a certain number of acquisitions, a new transition is requested due to the contrast between the new image (Figure 4.7 top left) and the current node. However, no similar past nodes are detected, and then a new node (Node 11) is generated and established as current node.

Despite the promising results obtained with the LexToMap technique, there are some failure cases as the one illustrated in Figure 4.8. It corresponds to the generation of the topological map from Sequence 1 Cloudy, and presents a node generation (Node 19) that should not have been performed, as there was a previous node (Node 13) created from images acquired in the same location. This failure may come from the threshold selection, which compromises a trade-off between specificity and generality, and aims to generate valid maps for all the sequences in the dataset. Another point to be taken into account is the difference of the corridor with respect to the rest of room categories. Namely, the corridor images in the dataset are unobstructed without the presence of objects. This avoids detecting some loop closures due to the lack of discriminating objects, as that shown in Figure 4.8.

The characteristics of the corridor also help us discover a proper behavior of the proposal: its adaptability to cope with heterogeneous rooms. Concretely, we can observe how largest transitions in the topological maps

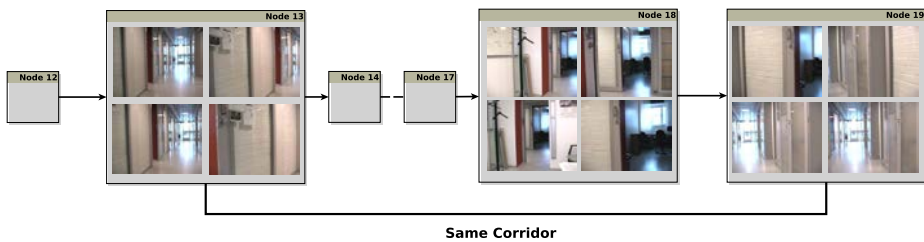


**Figure 4.7:** Transition generation during a LexToMap mapping procedure.

appear in the corridor area. This involves a lower density of nodes in this room category. This is desirable because the rest of rooms, especially the offices and the kitchen, are more suitable for incorporating relevant sub regions due to the presence of specific objects, like fridges or computers.

#### 4.5.4 Description Capabilities

The topological maps generated with our proposal present a clear advantage when compared with those generated with state-of-the-art approaches, namely their descriptive capabilities. This comes to the fact that each topological node is generated from a set of lexical labels that



**Figure 4.8:** Transition to an already visited location not detected.

can be used to describe its content. Figure 4.9 shows an example of a topological map generated from sequence 1 under cloudy conditions (Figure 4.6 top left). In this figure, we highlight a location (which belongs to the one-person office room category in the dataset), along with some of the images this location consists of, and the lexical labels word-cloud. This cloud is computed from the set of most-representative lexical labels, where font sizes denote the likelihood of the label in this location.

In addition to the descriptive capabilities, the lexical labels are amazingly useful for goal-driven navigation tasks. That is, the labels associated to a topological location refer to the content of the scenes, and therefore can determine the type of actions the robot would perform. In order to illustrate this capability, we remarked the locations on the same topological map including three different labels in their top-five most representative (higher likelihood) ones: desktop computer, refrigerator and banister.

This is shown in Figure 4.10, and it can be observed how all the locations selected with the label “desktop computer” belong either to a one-person or two-person office, which are the semantic categories (in comparison with kitchen, corridor or printer area) more likely to contain a desktop computer. A similar scenario was obtained with labels “refrigerator” and “banister”, which select locations that belong to kitchen and corridor categories respectively.

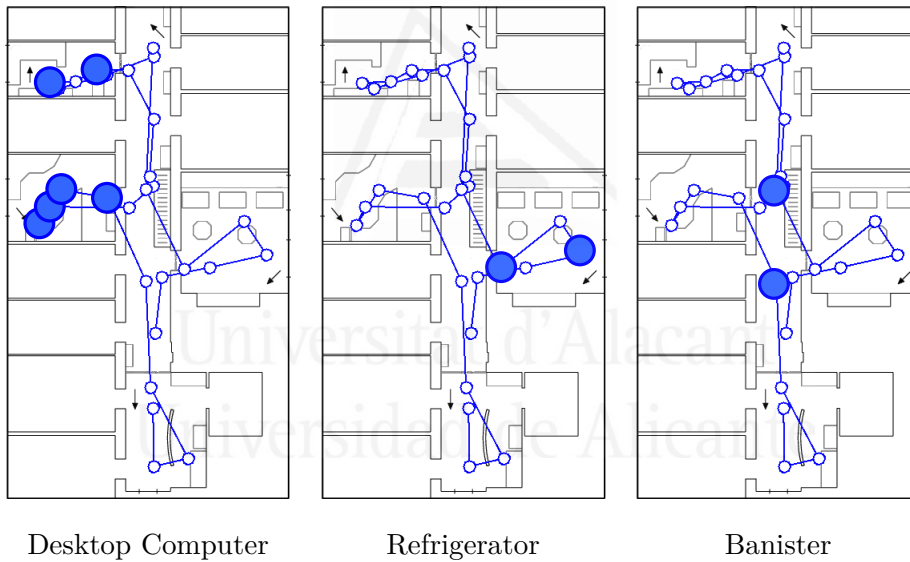
In addition, we are also interested in knowing how a lexical label is distributed over a topological map. For a better understanding we illustrated two examples in Figure 4.11 using the lexical labels “sliding door” and “photocopier”. In this figure, we use a heat color coding to represent the probability of describing each location using the provided lexical labels.

## 4.6 Conclusions and future work

This chapter has presented a novel approach for lexical-based topological mapping. The proposal relies on the annotation capabilities of the available pre-trained CNN models, and takes advantage of them to compute the similarity between input images. This strategy presents two main benefits. Firstly, the similarity between map locations is computed from

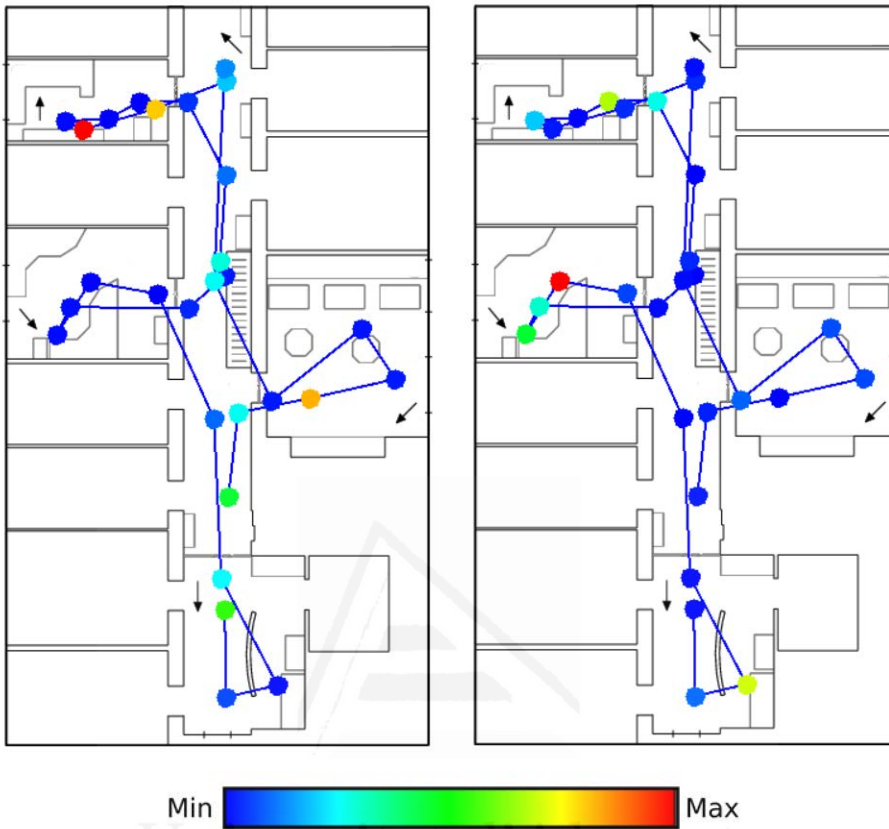


**Figure 4.9:** Location description by means of the cloud of representative lexical labels.



**Figure 4.10:** Locations representative of the lexical labels “desktop computer”, “refrigerator” and “banister”.

a lexical point of view and not only from visual feature similitude. This point increases the robustness of the method to challenging environments with small lighting variations or changes in the viewpoint. Secondly, the integration of annotations with lexical labels in the map generation procedure increases the representational capabilities of the maps, as locations can be described using a set of lexical labels. Moreover, these labels could



**Figure 4.11:** Topological maps where color codes represent the probability of describing each location with the lexical labels “sliding door” (left) and “photocopier” (right).

be extremely useful for future navigation purposes. Based on the results obtained under different lighting conditions, we can conclude that valuable topological maps can be obtained by following a standard approach without the need for selecting and tuning computer vision (for feature extraction) or machine learning (for matching and classification) algorithms.

We have presented a qualitative evaluation of our method. We know that a quantitative metric must be provided for a better evaluation. However, for the best of our knowledge, there is a lack of such quantitative metric and we plan to develop it in future work.

We also have in mind the comparison of the maps generated from



different proposals, including traditional approaches using visual features. To this end, as well as to automatically select the optimal values for the thresholds included in the algorithm, we are additionally working on the proposal of a metric suitable for evaluating the goodness of any topological map. In other future lines we propose to test the lexical descriptor in semantic scene classification and mapping, through unsupervised grouping techniques.



Universitat d'Alacant  
Universidad de Alicante

# AuSeMap: Automatic Semantic Map Generation

---

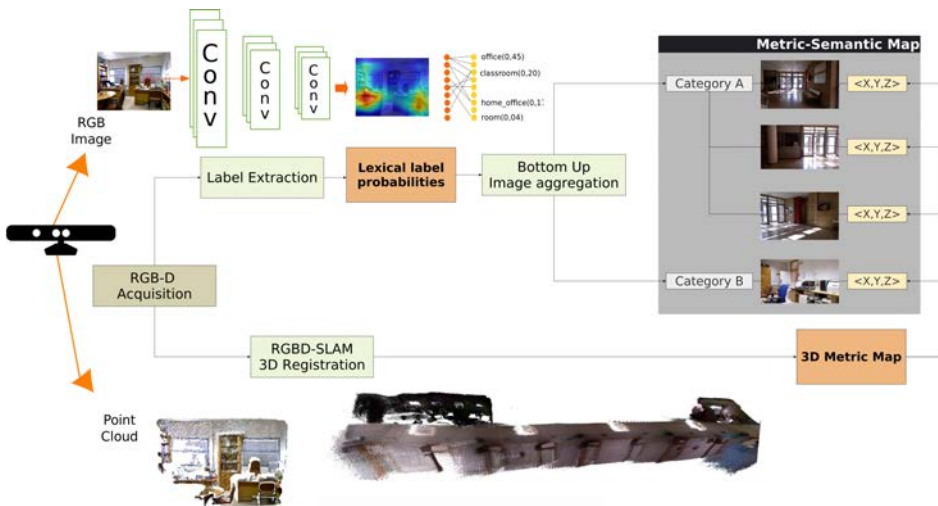
This Chapter shows the procedure developed to build a Semantic Map, based on a lexical descriptor that have been generated with CNN pre-trained models. Firstly, Section 5.1 presents an introduction to the semantic mapping problem. Then, Section 5.2 describes several works which have tried to solve existing mapping problems. Followed by Section 5.3 that shows the procedure to build a semantic descriptor through CNN trained models. After that, the proposal is deeply explained in Section 5.4. The experimentation and its results are detailed in Section 5.5. Finally, Section 5.6 outlines the main conclusions and topics for future work.

## 5.1 Introduction

Human supervision is still one of the major bottlenecks in the development of robotic systems. In addition to remote controlling or safety supervision, humans are usually required to annotate images or even complete sequences as a prior step to the generation of machine learning solutions. This step may also be needed to adapt generic solutions to more domain specific ones. Human supervision is also required to select the most appropriate environment representation, which is key to the problem of robot mapping and localization. Broadly speaking, we can find two

different families of representations in robotic mapping: metric and semantic. While metric maps refer to the positions the robot may reach in the environment, semantic representations incorporate additional information related to high-level descriptions of goals within the map. For instance, a semantic map can help to determine the expected behavior of a robot based on its current semantic localization (e.g. a kitchen, a bedroom, a living room). Even though semantic and metric localization can be carried out separately, the explicit association between metric locations and semantic labels, by means of a common map, has proven to be an efficient way to manage the metric semantic gap. In this chapter, we present a novel proposal to generate semantic maps automatically from sequences of unlabeled RGB-D robot acquisitions. The approach exploits the annotation capabilities provided by previously trained Convolutional Neural Network (CNNs) classification models, given input RGB images. From these annotations, we follow a bottom-up aggregation approach where similar RGB images are associated with the same semantic category. The depth (-D) information is used to obtain the global metric position of the robot by means of a RGB-D Simultaneous Localization and Mapping (SLAM) procedure. To this end, we draw on current and state-of-the-art approaches that have proven their validity in a range of scenarios. The overall scheme of the proposal is depicted in Figure 5.1, where the keystone of the proposal is the similarity computation between images by means of lexical annotations.

The proposal is validated using ViDRILO [Martínez-Gomez et al., 2015] (see Section 1.4.3), a dataset consisting of sequences of RGB-D images acquired by a mobile robot within an indoor environment. Images in the dataset are annotated with the semantic category of the scene where they were acquired. Therefore, we use this ground truth information to evaluate the appropriateness of our approach. The evaluation demonstrates that semantic maps can be successfully generated in a completely automatic way. Moreover, we have seen that the use of lexical annotations is not limited to generating semantic categories. These annotations also contribute to the descriptive capabilities of the generated maps. That is, each category can be described by means of its more representative lexical annotations,



**Figure 5.1:** Overall scheme of the proposal for generating semantic maps from lexical annotations.

which improves the interpretability of the environment representation.

## 5.2 Semantic Mapping

Map building or mapping is the process of generating an environment representation, namely a map, while a robot moves around the given environment [Thrun et al., 2002]. The end goal of the robot application determines the type of map to be selected, as we can find a range of maps including metric, topological or semantic representations. Metric maps integrate information about the location of obstacles (usually walls in indoor environments), and these locations are commonly exploited to automatically estimate the localization of the robot in further steps. The simultaneous problem of localizing a robot while it generates a map is known as SLAM, which has been extensively studied in the literature [Blanco et al., 2007, Burgard et al., 2009, Thrun and Leonard, 2008]. Topological maps (Section 4.2) describe the environment using a graph-based representation consisting of robot locations (nodes) and the set of available transitions between them (arcs). This type of map defines the paths the robot can

follow to reach a desired position, which is extremely useful for navigation purposes. While both metric and topological maps have traditionally been generated from laser sensors [Choset and Nagatani, 2001], there are several alternatives relying on visual images as the input source [Se et al., 2005, Lemaire et al., 2007]. However, these alternatives are not robust enough and far from perfect, so visual SLAM is still a challenging research area [Fuentes-Pacheco et al., 2012].

Meanwhile, several proposals have been developed to tackle semantic mapping, using different approaches to code images that represent the environment. A recent survey [Kostavelis and Gasteratos, 2015] exposes this considerable amount of works that focus on solving indoor large-scale interpretation problems, dealing with it by applying diverse techniques such as the Bayesian Reasoning used in [Liu and Von Wichert, 2013] that produces a generative model of the environment. [Galindo et al., 2005] employ the conceptual hierarchy to code the semantic knowledge about a problem involving two categories. [Wang and Lin, 2011] use the scaled CAD models of the environment and take advantage of the trajectory described by the robot to semantically annotate the visited areas. In [Rituerto et al., 2014], a catadioptric vision system is used to produce semantic-topological maps of indoor environments.

[Óscar Martínez Mozos et al., 2007] extracted geometric features from sensor range data of a classroom building, then trained an AdaBoost classifier to label images captured in a different building. [Kostavelis and Gasteratos, 2013] generated appearance based histograms to code each image, these histograms are built using visual words extracted from a visual vocabulary, and then used to train a SVM classifier. Based on the trajectory followed by the robot, a graph of nodes is generated taking into account a threshold distance between all nodes. Next, these nodes are labeled using a voting procedure of the SVM models. In this proposal the generation of nodes with similar labels leads to the definition of a zone such as “office” or “corridor”, but the labeling of several zones with the same semantic class is made by considering the distance between the zones.

[Pronobis et al., 2010] proposed a multi-modal semantic place classification, which focus on combining the use of local and global features of

an image with geometric features extracted from laser range data. Then, these features were used to train different SVM classifiers. At the classification stage, the features vector for every captured image was computed, and the resulting classes assigned for each one of the three features vector (local, global and laser) were sent to a SVM-DAS classifier, which would define the label for the image. The semantic zone annotation procedure was carried out by means of a confidence-based spatio-temporal accumulation method that took into account the predicted output for an image, the metric SLAM data, and the doors detection system to define the boundaries of the rooms.

Lately, the use of deep learning architectures such as Convolutional Neural Networks (CNNs) has been the focus of much attention as they have achieved excellent results in complex challenges such as the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [Krizhevsky et al., 2012]. Consequently, CNNs have become the *de facto* standard solution for visual categorization from large annotated sequences of images. Moreover, the results produced by the use of CNNs have motivated their use in semantic mapping problems. Most of the proposed methods use the CNN features extracted from one layer of the net (deep features), in order to represent an image. [Sharif Razavian et al., 2014] was one of the first proposals that used the CNN features produced by a layer of a CNN. This work presented a study for Visual Instance retrieval. They divided the image in a set of patches and calculated the features for every patch in order to classify an image. In [Chen et al., 2014], the authors extracted deep features from each layer of a CNN architecture, and then used this representation in a feature matching procedure to classify the input image. [Sünderhauf et al., 2015b] focused on the identification of landmarks in images, they applied a region selection procedure and then codified each region by means of their CNN features. [Sünderhauf et al., 2015a] developed a study for every layer of an AlexNet architecture. And, after the CNN features extraction, a nearest neighbor algorithm was applied to classify the images belonging to the test datasets.

Finally, the availability of pre-trained CNN models has opened up a new range of opportunities for exploiting high performance systems without

the storage and computing requirements traditionally needed to generate them. For instance, the output provided by CNN models pre-trained with a generic dataset, without any type of further tuning, was successfully used to perform scene classification in an indoor environment not previously imaged [Rangel et al., 2016a] presented in Chapter 3, as well as to build topological maps to represent an environment under several illumination conditions [Rangel et al., 2017] detailed in Chapter 4.

However, while these proposals are focused on the use of RGB images, the release of affordable RGB-D sensors, such as the Microsoft Kinect or the ASUS Xtion devices, has recently encouraged their use in a wide range of problems, like skeleton and gestures detection for human-robot interaction applications [Wu et al., 2012]. And, they have also been used for localization and mapping proposals with very promising results [Endres et al., 2012]. The use of RGB-D sensors for such specific goals avoids fitting robots with additional navigation-oriented sensors, as the visual camera integrated in the RGB-D sensor is usually required for many other tasks.

### 5.2.1 Place Recognition for Loop Closing

In relation to SLAM problems, solutions such as Real Time Appearance Based Mapping (RTAB-MAP) [Labbé and Michaud, 2011] focus on appearance-based loop closure detection for robots operating in large environments. This work employs visual characteristics to generate a signature for each acquired image, following a Bag of Words (BoW) approach. The aim of this proposal is to build a map in real-time, using only a selection of images from a sequence to detect a loop closure, but providing the option of knowing the coordinates of all the images in the sequence. The Continuous Appearance-Based Trajectory SLAM (CAT-SLAM) [Maddern et al., 2012] is a probabilistic approach mixing the spatial filtering characteristics of traditional SLAM with the FAB-MAP [Paul and Newman, 2010] appearance-based place recognition. The map generated is represented as a uninterrupted trajectory crossing all preceding locations. In order to detect a loop closure, a Rao-Blackwellized particle filter is used on several observations. Thus, the weight of particles is based on the local trajectory controlled by odometry and the likelihood of appearance-based

observations.

The FAB-MAP [Paul and Newman, 2010] approach uses BoW to determine whether an acquired image has been visited before or belongs to a new location. In addition to the visual processing of the scene, the aspects of its geometry are also taken into account. The FAB-MAP approach is based on learning a probabilistic model of the scene, employing a generative model of visual word observations, and a sensor model that explains unseen observations of visual words.

Using sensors to code features, DenseSLAM [Nieto et al., 2006] proposes a multi-layer representation approach. This proposes two ways of use layers. First, to use each layer to represent different environment's properties (traversability, occupancy or elevation). Second, each layer will represent the same feature of the environment but mean a different representation. The approach is able to generate dense maps without losing consistency.

Approaches such as Topological SLAM [Romero and Cazorla, 2010] relies on the matching of visual features, extracted from omnidirectional images that have been divided into regions, in order to build a topological maps grouping the extracted features in a graph.

Generation of metrics maps could be achieved using several proposals, such as the Real Time Camera Tracking and 3D Reconstruction, proposed by [Bylow et al., 2013], which uses a Signed Distance Function (SDF) to build a detailed representation of an indoor environment.

The BundleFusion method developed by [Dai et al., 2016] uses photometric and geometric matching of sparse features of a complete RGB-D images sequence, from a global set of camera poses. In order to build a robust pose estimation strategy. This strategy is optimized by frame and execute in real-time.

ElasticFusion [Whelan et al., 2016] combines a frame to model tracking and non-rigid deformation with surfel-based model data, to build a dense visual SLAM algorithm. This is made by developing several small loop closures detection with large scale loop closures for the existing and input models. It works by estimating the pose of the camera frame in the world frame using a blend of RGB alignment and ICP. The map is updated with



every new surfel added (with the camera pose), and the existing surfel information merges with the new one to refine their positions.

### 5.3 Lexical Labeling using CNNs

In this chapter, we are interested in the use of lexical annotations to automatically generate categories valid for robot semantic localization. This involves labeling each robot perception, namely a RGB-D image, with a set of lexical terms. This process consists of the use of previously-trained deep learning models, CNNs (see Section 1.5.1) in our case, to assign a probability value to each lexical label the model is able to recognize. This generates a lexical-based descriptor as represented by the Equation 4.2, that was described in the Section 4.3.1, of the LexToMap chapter.

Based on this representation, each robot perception is expected to semantically describe the place where the robot is located at that specific moment. Therefore, we can adopt the lexical encoding to automatically categorize the locations of any indoor environment from a semantic point of view instead of relying on visual appearance. That is, two visually different images would be assigned to the same category if they represent similar lexical labels. In order to use a specific set of lexical descriptors, we need to select the pre-trained CNN model which will define the lexical labels that will be used for representing the images.

### 5.4 Bottom-up Aggregation and Similarity Computation

The semantic mapping proposal follows a bottom-up aggregation process where sequences of RGB-D images serve as input. The depth information, in combination with the visual information, is used to estimate the pose of the robot corresponding to the acquisition. The visual information is exploited to extract a descriptor suitable for computing the similarity between two or more different robot poses. This is done by following the procedure previously presented, which relies on the lexical annotations provided by deep learning procedures.

The bottom-up aggregation is carried out by means of a hierarchical clustering approach. The starting snapshot establishes all the input images as individual clusters. Those clusters are then combined iteratively where most similar clusters are combined/aggregated in every new iteration. The process stops when the most similar clusters are different enough to be estimated as suitable to represent the same semantic category. This stopping condition can be tuned to select an optimal trade-off between specificity and generality. We discard the application of standard k-means clustering algorithms because the final number of clusters cannot be beforehand established. This algorithm requires a distance matrix for the whole set of images in order to initiate the clustering procedure. Then, hierarchical clustering uses this matrix as the information source to estimate the similarity between clusters and update it for every new cluster added to the set of clusters. This kind of algorithm employs a linkage strategy to select the pair of clusters that will be merged. In our cases we selected the Average Linkage Strategy that updates the matrix taking an average value of the distances between clusters.

Based on these preliminaries, the estimation of the cluster similarity is clearly established as the keystone of the proposal. As we adopt a representation where images are encoded using probabilities, we evaluate different similarity measures that have been extensively studied [Meshgi and Ishii, 2015]. Among all the available alternatives, including the  $L2$  distance, the cosine distance, the Pearson correlation or the Kullback-Leibler divergence, we selected the Kolmogorov-Smirnov distance ( $KS$ ), as it encodes the maximal discrepancy between two distributions. Hence, the  $KS$  distance is defined as:

$$KS(C_a, C_b) = \max_{1 \leq i \leq |\mathcal{L}|} |p_a(l_i) - p_b(l_i)| \quad (5.1)$$

where  $p_a(l_i)$  and  $p_b(l_i)$  denote the probability for the  $l_i$  in the clusters  $a$  and  $b$ .

The use of the  $KS$  distance helps us to discriminate between semantic categories by using those lexical probabilities that are more divergent. This is expected to hide small but numerous differences in lexical anno-

tations not corresponding to dissimilar semantic categories. Furthermore, the maximal discrepancy can help to identify lexical annotations corresponding to great changes in the appearance of the perceived scene. This may be due to the presence of objects not imaged in different aggregations of images.

---

**Algorithm 5.1:** Hierarchical clustering for bottom-up aggregation
 

---

```

input :  $\tau_d$  = Minimum distance to combine clusters
input : ImageList = Images acquired by the Robot
output: ClusterList =  $\emptyset$ 
1 forall image  $I_j$  in ImageList do
2   | Create a new cluster  $C_{new}$  from  $I_j$ 
3   | Add  $C_{new}$  to ClusterList
4 end forall
5 for every combination of clusters  $C_i, C_j$  in ClusterList do
6   |  $DistMatrix_{i,j}$  = KS distance between  $C_i$  and  $C_j$ 
7 end for
8 do
9   | Find a pair of clusters  $C_i, C_j$  in ClusterList that
10  |  $Dist_{i,j} = \underset{C_i \neq C_j; C_i, C_j \in ClusterList}{argmin} DistMatrix$ 
11  | CandidateClusters =  $\{C_i, C_j\}$ 
12  | if  $Dist_{i,j} < \tau_d$  then
13  |   | end While
14  | end if
15  | else
16  |   | Remove CandidateClusters from ClusterList
17  |   | Remove CandidateClusters from DistMatrix
18  |   |  $C_{comb}$  = combine CandidateClusters
19  |   | Add  $C_{comb}$  to ClusterList
20  |   | Update DistMatrix
21  | end if
22 while  $|ClusterList| > 1$ 
23 Return ClusterList

```

---

## 5.5 Experimentation and Results

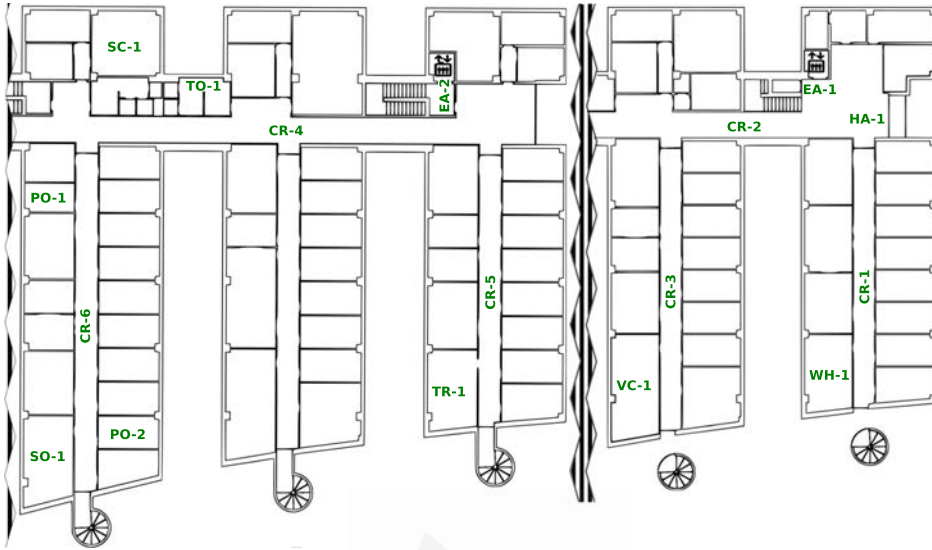
### 5.5.1 ViDRILO dataset

All the experiments included in this chapter were carried out using the ViDRILO (dataset[Martínez-Gomez et al., 2015]), see Section 1.4.3 for ad-

ditional details. From the whole dataset, we selected the first two sequences for the experimentation, which differ mainly in the acquisition procedure (clockwise and counter-clockwise). We supplemented the ground truth ViDRILO information with new classes that allow to distinguish between rooms belonging to the same category but located in different regions of the building. We can find this situation for the semantic categories “corridor”, “professor office” and “elevator area”. Table 5.1 shows the image distribution of this new classification, whereas Figure 5.2 shows the physical locations of the categories in the building. In order to clarify the difference between class and category, Figure 5.3 shows images belonging to different classes but sharing the same category. The visual dissimilarities between classes sharing the same semantic category can be observed there.

**Table 5.1:** Classification of ViDRILO images and distribution for sequences 1 and 2.

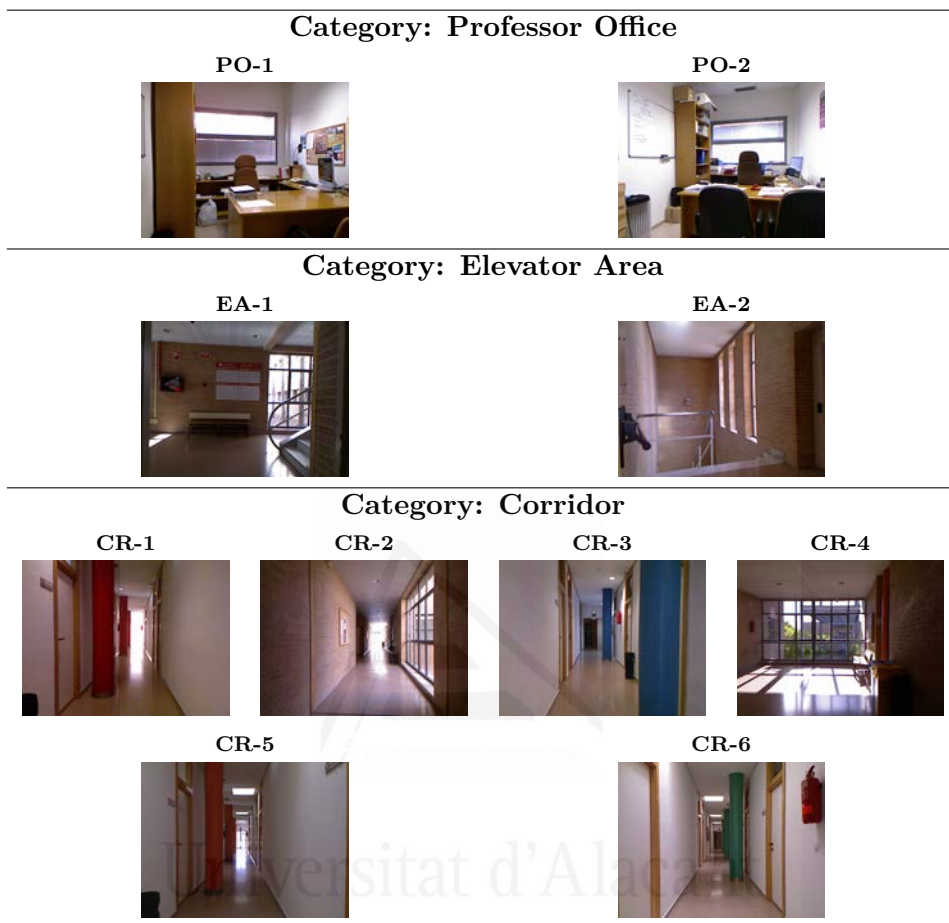
Class	Semantic Category	Sequence 1	Sequence 2
PR-1	ProfessorRoom	53	71
PR-2	ProfessorRoom	71	121
CR-1	Corridor	114	210
CR-2	Corridor	175	322
CR-3	Corridor	234	405
CR-4	Corridor	377	821
CR-5	Corridor	236	407
CR-6	Corridor	197	360
HA-1	HallEntrance	103	228
SR-1	StudentsRoom	155	276
TR-1	TechnicalRoom	136	281
TO-1	Toilet	121	242
SE-1	Secretary	98	195
VC-1	Videoconference	149	300
WH-1	Warehouse	70	166
EA-1	ElevatorArea	18	59
EA-2	ElevatorArea	82	115
<b>Total</b>		<b>2,389</b>	<b>4,579</b>



**Figure 5.2:** Localization of classes and ground truth categories in the ViDRILO Dataset.

### 5.5.2 Pre-trained models

The image descriptor generation process was developed using the Caffe Deep Learning framework, described in Section 1.6.2, due to the vast number of pre-trained models available and the ease of utilization of them. In order to generate the semantic descriptor for the sequences in the dataset, we have selected two popular architectures, namely GoogLeNet [Szegedy et al., 2014] and AlexNet ([Krizhevsky et al., 2012]), and two different datasets annotated with heterogeneous lexical labels: ImageNet and Places205. For additional details of the CNN architectures and datasets and models, please see Section 1.5.2, Section 1.4 and Section 1.6.2.1, respectively. Among all pre-trained models presented in Table 1.2 that could be suitable for the ViDRILO dataset we have selected the ImageNet-AlexNet, ImageNet-GoogLeNet, Places-AlexNet and Places-GoogLeNet. This selection is made taking into account the inner characteristics of the ViDRILO dataset, that represents a building environment, with several categories and objects whose lexical labels are present in the selected pre-trained models.



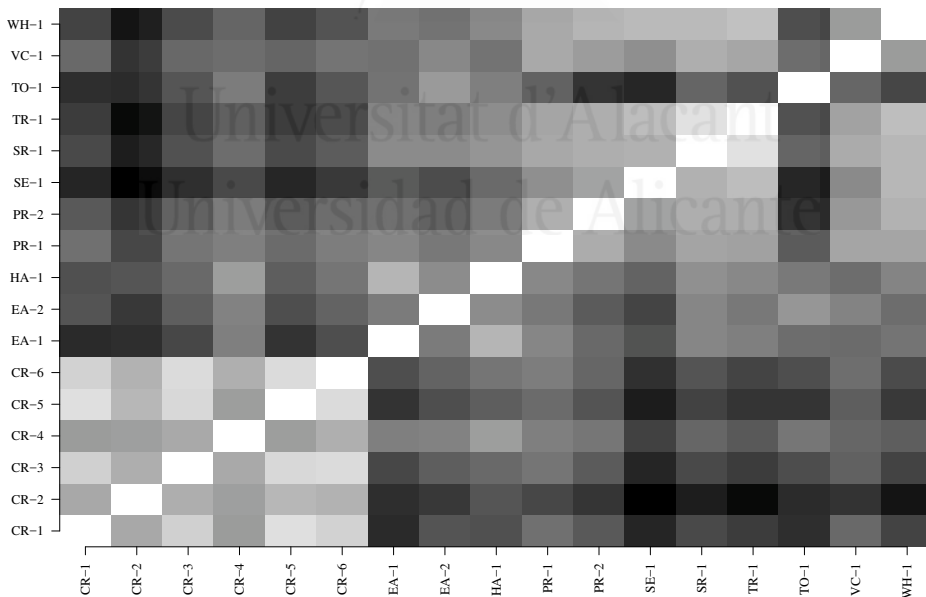
**Figure 5.3:** Comparative of images belonging to different locations with same category.

### 5.5.2.1 Baseline results

The initial distribution selected of the ViDRILO dataset was used to validate the appropriateness of the proposal as well as to highlight the challenges to be faced. To this end, we initially generated up to 17 aggregations of images per sequence attending to their category and physical location, as it was explained in Section 5.5.1. We then extracted the lexical annotations from the images using the four CNN models selected for the experimentation and introduced in Table 1.2. Next, using the Equation (5.1), we computed the Kolmogorov-Smirnov ( $KS$ ) distance between ev-

ery combination of aggregations, for instance between CR-1 & CR-1, and HA-1 & VC-1, among others. Finally, we generated the distance matrix shown in Figure 5.4. This matrix averages the distance between every pair of classes from the eight combinations of four models and two input sequences.

Some initial conclusions can be drawn from this preliminary experiment. Firstly, we can observe how the *KS* distance computed from the lexical annotations provided by the CNNs is suitable to determine the semantic appearance between different images. This arises from the fact that images sharing the same semantic category but acquired from distant places were found to be similar. The experiment also reveals that some partitions between semantic categories are artificial from a similarity point of view. These partitions may come from expected human behaviors, such as the discrimination between secretary office and professor room.



**Figure 5.4:** Distance Matrix for the categories in ViDRILO dataset.

### 5.5.3 Automatic semantic map generation

Using the two first sequences of ViDRILO as input, we evaluated the generation of semantic maps by following our proposal. This proposal consists of a bottom-up clustering approach as described in Algorithm 5.1, and relies on the image descriptors computed from the output of pre-trained CNN models. The process continues until the smallest  $KS$  distance between two candidate clusters, namely image aggregations, is above a certain threshold  $\tau_d$ . Hence,  $\tau_d$  will determine the number of generated clusters or semantic areas, being  $\tau_d \propto \frac{1}{\#clusters}$ .

During the experimentation, we evaluated five threshold values (0.2, 0.35, 0.5, 0.65, and 0.8) and four different CNN models (ImageNet-AlexNet, ImageNet-GoogLeNet, Places-AlexNet and Places-GoogLeNet). Table 5.2 shows the number of generated clusters for each combination of parameters. In the light of this information, we were able to obtain fine grain semantic distributions of the environment by selecting smaller  $\tau_d$  values. Aiming to exploit the annotations provided with the ViDRILO dataset, which considers 10 different semantic categories, we select  $T_d = 0.5$  for the rest of the experimentation. This value generates a reasonable number of clusters between 9 and 19.

The evaluation of a semantic mapping algorithm is not trivial, as it would depend on the subjective criteria and/or the final purpose of the environment representation. To this end, we propose to quantitatively measure the condition of the generated maps by taking advantage of the ground truth information provided with ViDRILO. This is done by studying the correlation between the clusters generated and the ViDRILO se-

**Table 5.2:** Number of clusters generated for different  $\tau_d$  and CNN combinations.

$\tau_d$	0.20	0.35	0.50	0.65	0.80
<i>Sequence</i>	<b>S1 S2</b>	<b>S1 S2</b>	<b>S1 S2</b>	<b>S1 S2</b>	<b>S1 S2</b>
ImageNet-Alexnet	110 109	30 32	13 12	6 4	2 1
ImageNet-GoogLeNet	134 175	40 43	19 20	12 10	3 4
Places-AlexNet	95 105	23 26	12 13	4 5	1 2
Places-GoogleNet	79 102	22 26	9 14	4 4	2 2



mantic categories. Drawing on the classification previously presented (see Table 1.1), the distribution of clusters for two different classes belonging to the same semantic category (e.g. CR-1 and CR-2) should be similar. This quantitative evaluation complements qualitative evaluation in the form of the visualization of the generated maps.

Based on the initial class distribution, we can generate up to 17 combinations of classes belonging to the same semantic category. This results from all combinations of CR- $a$  with CR- $b$  ( $a, b \in \{1, \dots, 6\} \wedge a \neq b$ ), EA-1 with EA-2, and PO-1 with PO-2. For each pair of these combined classes, we compute the  $L2$  distance of their cluster distributions. Hence, we define a metric value  $M$  to measure a given generated map. For each category  $n$  (e.g. corridor), we have  $|cat_n|$  classes (e.g. 6 in corridor).  $M$  is defined as the average distance between classes in the same category, for all the categories. The  $M$  formulation is presented in Equation (5.2), where  $cat_{n,i}$  denotes all the images belonging to the  $i_{th}$  class of the category  $n$ . This comparison process yields a distance metric in the range  $[0, 1]$  (normalization has been carried out by taking into account the dimensionality of the distributions). A graphical representation of this process for classes CR-1 and CR-2 is shown in Figure 5.5.

$$M = \frac{\sum_{n=1}^N \sum_{i=1}^{|cat_n|-1} \sum_{j=i+1}^{|cat_n|} L2(cat_{n,i}, cat_{n,j})}{\sum_{n=1}^N \frac{|cat_n|(|cat_n|-1)}{2}} \quad (5.2)$$

Table 5.3 shows the results obtained, where the most significant distribution variations were found for classes EA-1 and EA-2. This divergence is a result of the ViDRILO acquisition procedure: classes EA-1 and EA-2 are not only visually different, but they also differ in the nature of their content. That is, while EA-2 images perceive the elevator and the small region located on the first floor near the elevator, EA-1 images represent the building hall from the perspective of the area near the elevator on the ground floor. The cluster distribution was more uniform for the combination of classes belonging to the category corridor, especially for those not including classes CR-2 or CR-4 (these corridors integrate the stairs connecting the two building floors). Similar conclusions could also be extracted from the distance matrix computed in the baseline results (see Fig-

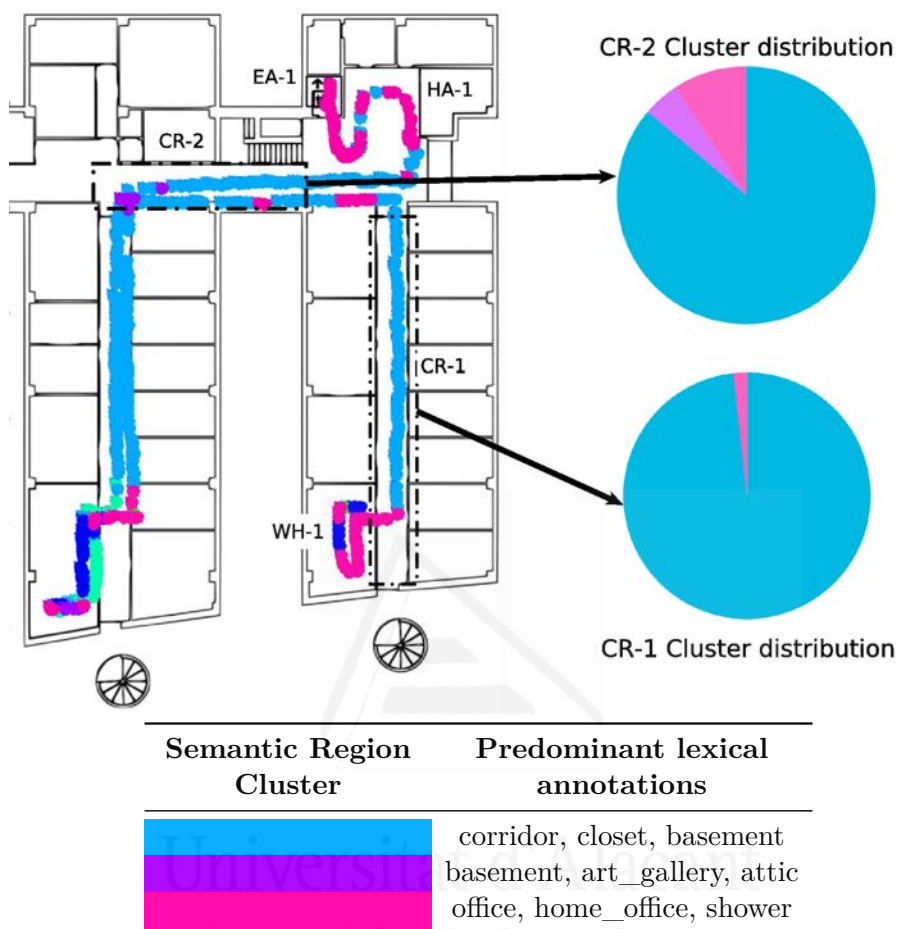


Figure 5.5: Quantitative metric generation.

ure 5.4). Based on the results obtained, we established Places-GoogLeNet as the optimal CNN model for the automatic generation of semantic maps.

#### 5.5.4 Analysis of maps generated

The semantic maps generated with our proposal present some novelties to be analyzed. Firstly, the maps are built in an unsupervised fashion once the combination of CNN model and threshold  $\tau_d$  has been set. As a result of the procedure, we obtain a set of clusters corresponding to semantic areas in the environment. All of them integrate information related to

**Table 5.3:** Distances in the cluster distributions for each combination of classes sharing semantic category. Results are averaged over sequences 1 and 2.

	ImageNet- AlexNet	ImageNet- GoogLeNet	Places- AlexNet	Places- GoogLeNet
<b>CR-1.CR-2</b>	0.56	0.32	0.06	0.11
<b>CR-1.CR-3</b>	0.13	0.16	0.21	0.13
<b>CR-1.CR-4</b>	0.15	0.37	0.41	0.40
<b>CR-1.CR-5</b>	0.15	0.17	0.08	0.11
<b>CR-1.CR-6</b>	0.10	0.17	0.20	0.14
<b>CR-2.CR-3</b>	0.64	0.22	0.18	0.07
<b>CR-2.CR-4</b>	0.49	0.30	0.38	0.37
<b>CR-2.CR-5</b>	0.54	0.38	0.04	0.09
<b>CR-2.CR-6</b>	0.61	0.35	0.16	0.10
<b>CR-3.CR-4</b>	0.22	0.33	0.28	0.42
<b>CR-3.CR-5</b>	0.16	0.19	0.16	0.05
<b>CR-3.CR-6</b>	0.12	0.15	0.10	0.12
<b>CR-4.CR-5</b>	0.25	0.39	0.42	0.44
<b>CR-4.CR-6</b>	0.19	0.33	0.30	0.33
<b>CR-5.CR-6</b>	0.09	0.15	0.16	0.13
<b>EA-1.EA-2</b>	0.42	0.77	0.79	0.44
<b>PO-1.PO-2</b>	0.38	0.43	0.32	0.45
<i>M<sub>value</sub></i>	<b>0.31</b>	<b>0.31</b>	<b>0.25</b>	<b>0.23</b>

the most suitable lexical terms to be found in such areas. These lexical annotations can be used to describe each region, but also to discover some particularities of the environment. Moreover, any agent operating in such regions may consider the lexical terms to adapt or modify its behavior.

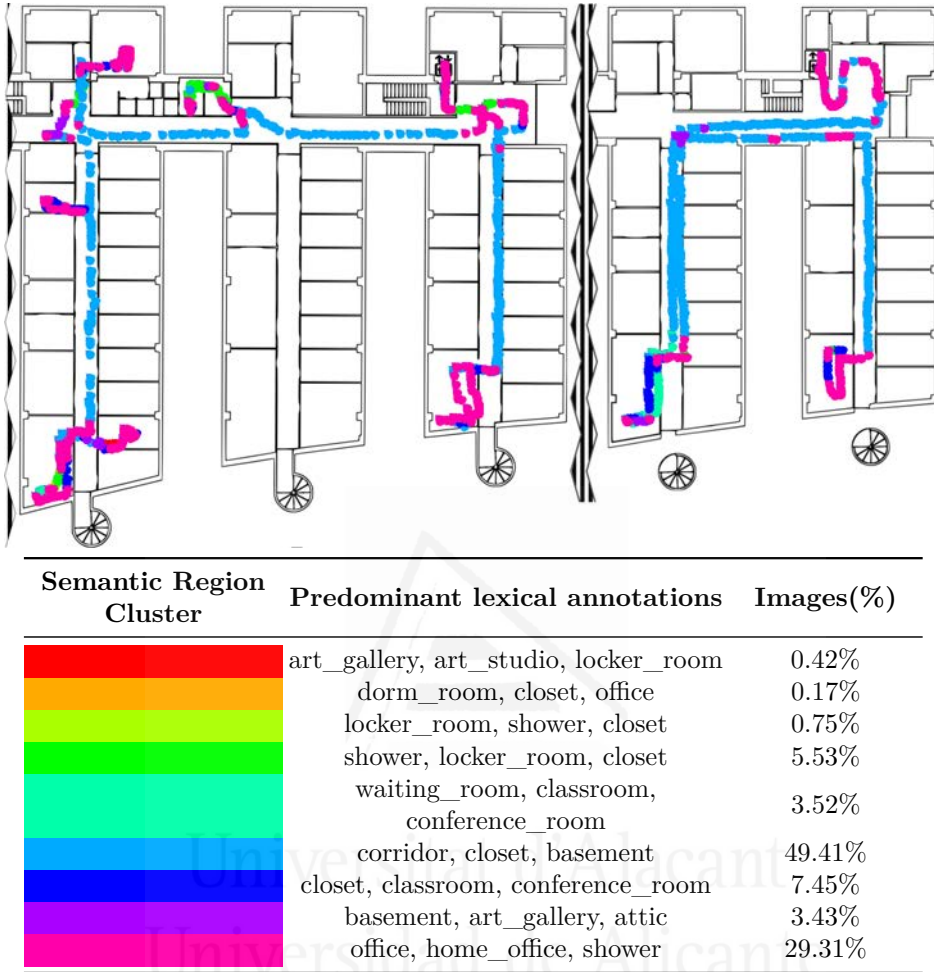
On the basis of the model and threshold combination selected in previous experiments, we generated two semantic maps from ViDRILO sequences 1 and 2, which are shown in Figure 5.6 and Figure 5.7 respectively. An initial view of these maps exhibits a main cluster incorporating a great percentage of the images. The biggest cluster in both maps consists of most of the images belonging to the category Corridor. In addition to this, the cluster is described using its predominant lexical terms, which also include “corridor”. Another relevant cluster incorporating a large number of images is related to a subset of semantic categories, like the aggregation of Professor Office, Student Office, Secretary, and Technical Room (the pre-

dominant terms for this cluster include “office”). It is also worth noting how a large percentage of the images acquired in the Toilet were associated with a single cluster. This cluster, in both sequences, had “shower” within its predominant lexical terms. On the other hand, these maps show that images from nearby locations may be associated with different clusters. This is due to the fact that: (a) the robot location and temporal continuity of the sequence had not been integrated in the procedure; and (b) the orientation of the robot may drastically change the portion of environment to be perceived. The robot position is not exploited to avoid negative effect due to errors on the registration process. Indeed, the use of the sequence continuity would increase the dependency on the acquisition procedure.

## 5.6 Conclusions and Future Work

In this chapter, we present a novel proposal for the automatic generation of semantic maps. The generation procedure starts from a sequence of RGB-D images acquired in the environment, and delegates the feature extraction to CNN models previously trained with lexical annotations. This provides a semantic point for the subsequent bottom-up aggregation process, where similar images are merged into clusters defining semantic regions or areas of the environment. The metric position associated to each input image is obtained by exploiting RGB-D SLAM solutions. The evaluation of the method has been carried out using ViDRILO, a dataset containing RGB-D images acquired from an indoor environment using a mobile robot.

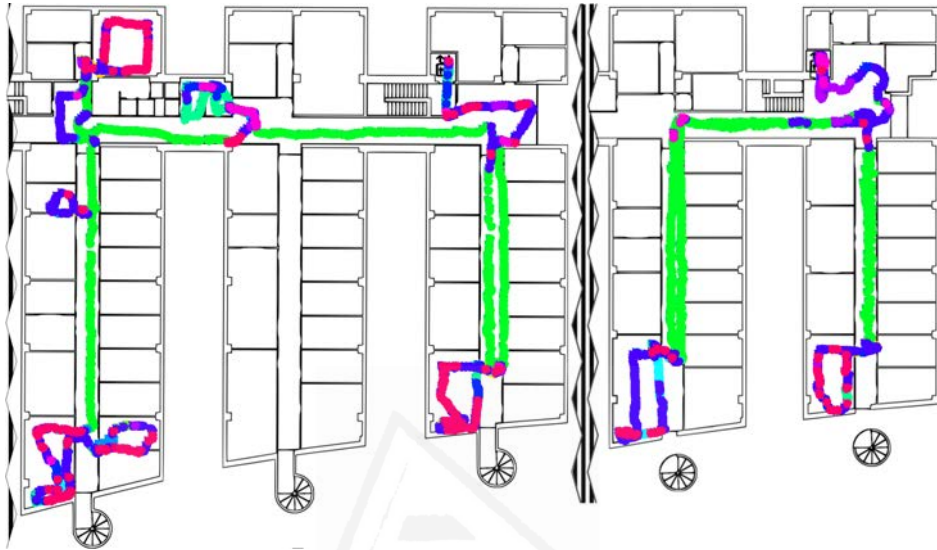
Some combinations of CNN architectures and datasets used to generate them have been tested and evaluated in this proposal. We have also discussed the effect of the  $\tau_d$  threshold parameter. Namely, we can adapt this value to generate semantic representations with different degrees of granularity. In view of the results obtained, we can initially conclude that suitable semantic representations can be automatically generated thanks to our proposal. Despite the difficulty in objectively evaluating the quality of the generated maps, we exploit the ground truth information provided with ViDRILO to validate the appropriateness of the maps generated.



**Figure 5.6:** Semantic map generated from Sequence 1 with the Places-GoogLeNet model.

Based on the maps generated, we also conclude that most of the semantic environment partitions performed by humans actually correspond to fuzzy expected behaviors, instead of justified divergences in their content.

As future work, we propose to integrate the information provided by the semantic regions with the inference process. To this end, a mobile robot would determine its most appropriate behavior (when available) based on the lexical annotations included in the region where it is located. This advanced reasoning will also be incorporated in planning strategies.



Semantic Region Cluster	Predominant lexical annotations	Images(%)
	jail_cell, locker_room, closet	0.09%
	reception, office, home_office	0.13%
	closet, basement, shower	2.66%
	kitchenette, kitchen, reception	0.09%
	bookstore, closet, office	0.17%
	corridor, lobby, basement	40.85%
	shower, basement, locker_room	5.90%
	waiting_room, conference_room, classroom	2.49%
	art_gallery, art_studio, corridor	1.79%
	shower, locker_room, closet	3.28%
	classroom, office, corridor	27.09%
	staircase, sky, basement	1.88%
	attic, basement, corridor	2.32%
	office, home_office, classroom	11.27%

**Figure 5.7:** Semantic map generated from Sequence 2 with the Places-GoogLeNet model.



Universitat d'Alacant  
Universidad de Alicante

# OReSLab: 3D Object Recognition through CNN Semi-Supervised Labeling

---

This Chapter describes the processes to build a 3D object classifier using clusters of points that have been labeled using an external deep learning-based labeling tool. The chapter includes the validation of the proposal using a set of different objects and combination of these. Firstly, Section 6.1 defines the problem and exposes the difficulties faced in order to achieve a robust object recognition method. Next, Section 6.2 details state-of-the-art related to object recognition. Following, in Section 6.3, the details of the proposed pipeline are presented. In Section 6.4, we describe how the experimental set was selected, and we present the experimental results obtained. Section 6.5 presents a discussion of the experimental results. Finally, in Section 6.6 the main conclusions of this chapter are outlined.

## 6.1 Introduction

Object recognition is a challenging research area of growing interest in recent years due to its applicability in fields such as autonomous robotics and scene understanding. The research in this field has been stimulated by the appearance of more sophisticated cameras, as well as the capability



of learning from a vast amount of images.

Nowadays, we can clearly differentiate between object recognition approaches based on 3D or 2D image information. Conventional 3D object recognition approaches [Tombari et al., 2011, Asari et al., 2014, Aldoma et al., 2012, Rangel et al., 2016b] deal with problems such as occlusions, holes, noise, and rotation, translation or scale invariance. The use of 3D information is computationally expensive, and demands enormous storage facilities. This last point, in conjunction with the expense of manually labeling 3D images, makes it difficult to release interestingly large object recognition datasets with 3D labeled objects. The lack of datasets with sufficient 3D information of a vast range of objects represents a persisting problem, as some of the most outstanding machine learning techniques, especially Deep Learning (DL), require enormous labeled sequences for their effective training. On the other hand, 2D object recognition has benefited from the release of Deep learning techniques, and more specifically Convolutional Neural Networks (CNNs), to obtain promising results in recent years [Bui et al., 2017, Girshick, 2015, Girshick et al., 2014, Sermanet et al., 2013, Ren et al., 2015a]. CNNs trained from huge datasets like ImageNet present high generalization capabilities using a varied kind of objects and heterogeneous images.

Despite the latest advances in 2D object recognition, there are still several drawbacks to the use of perspective images, such as the difficulty of working in dark environments. This point may be a requirement for most robotic applications like vigilance or inspection. Moreover, the geometry of the object may be relevant to avoid false positives due to the presence of pictures in the scenes. That is, 2D object recognizers can wrongly recognize objects if they are provided with input images containing pictures of such objects. For instance, a 2D fruit recognizer can wrongly recognize an apple if presented with an image from a scene including some food advertisement posters.

The work presented in this chapter proposes to exploit the experience in both 2D and 3D object recognition to generate solutions in a semi-supervised fashion. To this end, we exploit the labeling capacities of DL-based current 2D solutions [Deng et al., 2009, Russakovsky et al., 2015, Jia

et al., 2014], but relying on 3D input images to avoid the problems already described. Figure 6.1 presents our approach graphically, which is also described below.

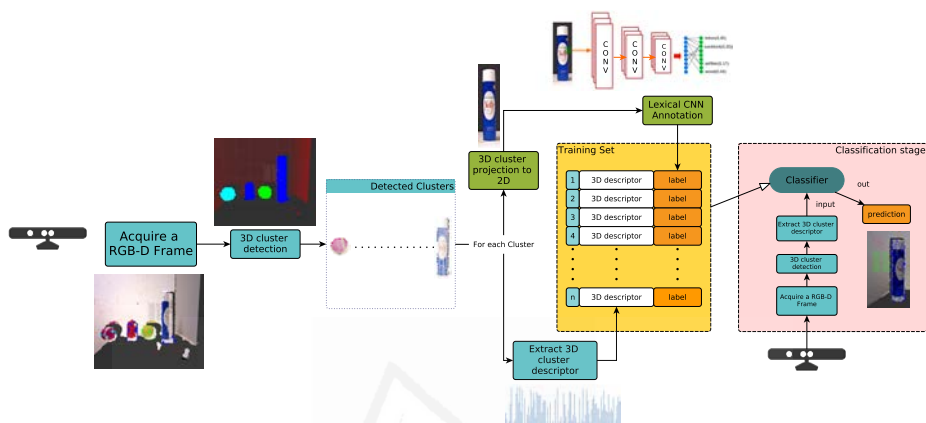


Figure 6.1: Proposal flowchart.

The proposal initially detects objects in 3D images, encoded as point clouds, and segmented using a clustering algorithm. Each cluster is projected into a perspective image, which will be labeled employing a black box system, as most of the internal parameters of the labeling procedure do not need to be known. Initial labeling is carried out by a CNN model previously trained from any of the huge 2D datasets annotated with object information. There are several CNN models freely available and capable of providing information for hundreds of different object categories. In a parallel way, the 3D data from each cluster are processed to extract a 3D descriptor. The set of descriptor- label tuples will serve to train a classification model, which will carry out the effective object recognition process.

Our proposal makes the most of mixing the procedures of 2D data that feedback the output of 3D procedures. This is due to the employment of 3D clustering procedures that help us obtain better segmentation of the objects present in the scene. Hence, the interplay of the two kinds of data is a crucial milestone of the proposed approach. As a consequence of relying

on an external labeling tool to generate the training set, some images may be wrongly classified, and the further learning stage of the proposal must be robust enough to cope with flaws in the training data. Hence, the main contributions of the chapter can be outlined as follows:

- Exploiting the knowledge included in CNN models previously trained to annotate 3D input images.
- Training a sufficiently robust classification model to be generated from noisy training sets.

## 6.2 Related Work

There exist several approaches to performing 3D object recognition, most of them based on feature matching. One example is presented in the review by [Guo et al., 2014], where the authors detail the basic stages for the recognition, and also describe the available datasets, feature descriptors, and keypoint detectors that have been widely used in a vast majority of the studies. Another example is [Tombari et al., 2011] where the authors study the recognition ability of stereo algorithms. Similarly, the solution presented in [Asari et al., 2014] evaluates several 3D shape descriptors to determine their feasibility in 3D object recognition tasks.

Using a combination of images and depth maps for cluttered scenes, as proposed by [Hinterstoisser et al., 2011], the authors obtain positive recognition results. In order to detect free-form shapes in a 3D space, a Hough Voting algorithm was proposed by [Tombari and Di Stefano, 2010] yielding good recognition rates. [Pang and Neumann, 2013] describe a general-purpose 3D object recognition framework. This framework works by combining 3D local features with machine learning techniques in order to detect objects in different kinds of images, such as engineering scenes or in street captures. Using a novel Global Hypothesis Verification algorithm, [Aldoma et al., 2012] refine the obtained results, discarding false positive instances. Based on the former work, [Rangel et al., 2016b] carried out a study aimed at testing whether a Growing Neural Gas (GNG) could reduce the noise in scene point clouds and therefore manage to recognize

the objects present in the 3D scene.

In recent years, DL has been widely applied to solve the 3D object recognition problem. Different approaches have been taken, but with promising results, even though these models tend to be slow for training. However, 3D object recognition challenges like ModelNet <sup>1</sup> aim to find a DL-based approach to produce an accurate classification model in the challenge dataset. The challenge provides participants with a dataset with 10 or 40 categories of CAD objects. Solutions presented for the challenge take diverse approaches to tackle the problem. [Wu et al., 2015] propose to represent 3D shapes as probabilistic distributions using binary variables on a grid of 3D voxels. This work also introduces the ModelNet dataset used in the challenge.

Nowadays, a trend in recognition is to mix several representations and CNN design in order to generate sufficiently discriminative information about the objects. For instance the DeepPano approach presented by [Shi et al., 2015] uses a cylinder projection in the principal axes of the object to build a panoramic view. This view will be used for learning the representation by a modified CNN. The solution presented by [Wu et al., 2016] proposes employing a 3D Generative Adversarial Network (GAN) together with a 3D CNN to capture 3D shape descriptors, and then uses it in classification tasks.

The VoxNet [Maturana and Scherer, 2015] proposal also employs a 3D CNN, but mixing it with a representation on 3D occupancy grids to efficiently recognize 3D shapes. The PointNet [Garcia-Garcia et al., 2016] approach employs 3D Convolutions for extracting object features represented by density occupancy grids and constructs a 3D classification model.

Utilizing the GPU acceleration, [Bai et al., 2016] propose a 3D Shape Matching procedure named GPU acceleration and Inverted File Twice (GIFT). GIFT uses a combination of projective images of 3D shapes using features extracted with a CNN. These features are then matched and ranked to generate a candidate list of labels. The work presented by [Johns et al., 2016] proposed the use of a sequence of multiples views of an object.. Next, the views are grouped in pairs and then, a CNN is used to classify

---

<sup>1</sup><http://modelnet.cs.princeton.edu/>

the pair. After re-classification, the contribution of each pair is weighted for learning an object classifier.

In FusionNet [Hegde and Zadeh, 2016] a pair of volumetric CNN and a Multiview Convolutional Neural Network (MVCNN) are fed using a combination of Pixel representation (image projection), and volumetric representation (voxel grid) producing good recognition results. Similarly, the MVCNN has been used by [Su et al., 2015], employing a rendered collection of image views to learn to classify objects.

The work put forward by [Sinha et al., 2016] proposes the creation of geometric images through the mapping of the surface of a mesh in a spherical parametrization map, projects them to an octahedron, and then forms a square after a cut and ensemble operation.

The Voxception ResNet (VRN) proposed by [Brock et al., 2016] is a ResNet-based architecture which works by concatenating standard ResNet blocks to produce inception modules. The VRN is fed by a voxelized volumetric input and then their predictions are summed to generate an output.

The LonchaNet technique, proposed by [Gomez-Donoso et al., 2017], focuses on solving the object classification by using slices of the views of a 3D object. Then, using an ensemble of GoogLeNet architectures, it learns the features of the objects.

The use of CNN for object classification has introduced several advantages in the field. These advantages may be resumed as follows. Firstly, the existence of several DL frameworks and the ease of use of these allows the users to straightforwardly develop their own models based on their specific requirements. Among these frameworks, we can mention TensorFlow [Abadi et al., 2015]<sup>2</sup>, Theano<sup>3</sup>, MxNet<sup>4</sup> and Caffe [Jia et al., 2014]<sup>5</sup>. Secondly, owing to these frameworks, the user community has created a massive library of models that others can freely use. These models were also trained with different image datasets, such as [Deng et al., 2009], Places [Zhou et al., 2014] or a combination of both. With this in mind,

---

<sup>2</sup><https://www.tensorflow.org/>

<sup>3</sup><http://deeplearning.net/software/theano/>

<sup>4</sup><http://mxnet.io/>

<sup>5</sup><http://caffe.berkeleyvision.org/>

we find another leverage point, the generalization capability, as a consequence of the use of the CNN to learn features from a vast number of images. Moreover, the use of these pre-trained models avoids having an enormous amount of data (dataset of images) in order to train a classifier. Consequently, applications demand less storage space and processing time.

The CNN has been used in object detection tasks and challenges such as the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [Deng et al., 2009, Russakovsky et al., 2015]. This challenge focuses on processing 2D images in order to achieve Scene Classification and object detection in the ImageNet dataset. In the latest editions, winning teams have based their solutions on the use of CNNs.

Related to the 2D object recognition, the proposal of [Uijlings et al., 2013] is based on a combination of exhaustive search and image segmentation to recognize possible object locations in a scene, then using Bag-of-Words with a SVM for object recognition. The Regions with CNN Features (R-CNN) presented by [Girshick et al., 2014] proposes the use of a preliminary phase of region detection. Next, it computes the features of the wrapped zone with a CNN and uses an SVM linear classifier to detect the objects in the regions. This approach has been tested in the 200-Class ILSVRC2013 dataset. Based on this approach, the solution presented by [Ren et al., 2015b] introduces the Region Proposal Network (RPN), which generates region proposals, which will be used for a R-CNN in object recognition tasks. Both networks can be trained to share convolutional features.

In the work proposed by [Bui et al., 2017], an AlexNet CNN architecture is modified by replacing the fully connected layers with a Recursive Neural Network in order to process the features extracted by the convolutional architecture. The approach has been tested for detecting objects present in the images.

CNNs have been also used as supervised object detectors such as those presented in [Sermanet et al., 2013, Ren et al., 2015a, Girshick et al., 2014, Girshick, 2015], but these detectors require the bounding box annotation for training the models. Meanwhile, the proposal by [Tang et al., 2016] focuses on developing object detectors by transference of visual and semantic knowledge. These detectors are derived from image classifiers

and are constructed taking into account the visual and semantic similarities between the target object and the kind of images used for training the classifier.

Others approaches, such as [Wang et al., 2015, Song et al., 2014, Bilen et al., 2015, Bilen et al., 2014], involving the use of CNN for object detection, are known as weakly supervised methods. These methods train a CNN detector using images that have been labeled at image level and without the bounding box information of the objects. Therefore, these approaches employ the CNN as a feature extractor.



In the work presented by [Qu et al., 2017], the authors focus on detecting salient objects in an image by using RGB and depth images. The authors used several saliency feature detectors and fed a CNN to output the probability that a superpixel belongs to a salient area of the image. Finally, the saliency map is generated using a Laplacian Propagation with the output of CNN.

Although the previous 2D object recognition approaches provide good classification results, there are still some problems to be solved. For example, these approaches will provide the same output for an image of a given object and for an image of a picture of such object. Table 6.1 shows the problem graphically. The left-hand column of the table corresponds to a picture of a cat accurately classified by the labeling tool, whereas the right column shows a picture taken of the same picture, but displayed on a laptop screen. Both images were labeled as *Egyptian cat* despite the second corresponding to a picture of another picture.

### 6.3 OReSLab: 3D Object Recognition through CNN Semi-Supervised Labeling

The aim of our proposal is to build an object recognizer based on 3D descriptors, but generated in a semi-supervised way. To this end, our technique generates a training set trusting in the labels provided by an external agent, namely a CNN previously trained with large quantities of perspective images. In addition to the labels, we use 3D fixed-dimensionality features extracted from input point clouds, aiming to combine procedures

**Table 6.1:** Comparison of the labels produced for two version of same picture of a cat.

Original Image		Picture	
			
Predicted Classes			
Label	Probability	Label	Probability
<u>Egyptian cat</u>	0.59773898	<u>Egyptian cat</u>	0.18463854
<u>tabby</u>	0.28014019	screen	0.12349655
tiger cat	0.07507683	<u>tabby</u>	0.10294379
lynx	0.02829022	monitor	0.09576164
wood rabbit	0.00240113	television	0.04949423
wallaby	0.00170649	laptop	0.03861517
Siamese cat	0.00127562	binder	0.03058814
triceratops	0.00126094	home theater	0.02762092
wild boar	0.00121317	Scotch terrier	0.02489605
grey fox	0.00092880	notebook	0.01437293

from 2D a 3D scopes. The training set would include a new instance for every cluster detected in the input point cloud, as they can feasibly correspond with an object.

OReSLab involves a training stage from the previously generated training set. We should take into account that the labeling system may initially present classification errors. Therefore, we assume our classification model must deal with errors derived from the labeling tool. Figure 6.1 shows an overall flowchart for the proposal. Green boxes indicate processing of an RGB (2D) image, whereas the blue ones indicate processing of a RGB-D (3D) point cloud. The solution presented in this chapter can be divided in four different phases, namely Acquisition, 2D Labeling, 3D Description, and Training and Validation.



### 6.3.1 Acquisition Phase

This phase begins with the acquisition of the RGB-D data or point cloud. This point cloud is then processed using a Euclidean clustering algorithm in order to detect point clusters. Detected clusters will be treated as objects and then stored to be used in following phases. Each object instance is captured in several vantage points until a defined number of samples are achieved. Figure 6.2 (left) shows the input cloud of this phase, with the respective image of the cloud. The figure also shows the detected clusters (center) in this input cloud.

### 6.3.2 2D Labeling Phase

Once clusters have been identified, every one is projected into a 2D RGB image. Then, the next step is to tag this image with a representative lexical label. The tag will be assigned by an external labeling tool, a CNN in our proposal. This tool could be a deep learning classifier model, a human expert or any system capable of generating a set of lexical labels with a confidence value as output. Figure 6.2 right shows the projected images belonging to the clusters detected in the acquisition phase, each projected image will be labeled using the labeling tool.

### 6.3.3 3D Description Phase

The description phase takes each identified cluster in the point cloud and then proceeds to compute a 3D global descriptor from all of them. Next, this descriptor is stored for posterior uses. In the same way the 2D labeling phase, this phase is performed for every cluster detected in the original point cloud.

### 6.3.4 Training and Validation Phase

Once the above phases have finished, the next step is to train a classifier using every sample of the object instances, in addition to the respective label assigned by the labeling tool. This classifier is then validated using a baseline scenario that will be described in the next section.



**Figure 6.2:** Example of the Acquisition (left) and 2D Labeling Phase (right) of the proposal.

## 6.4 Experimentation

### 6.4.1 Object Selection for experimentation

In order to validate the proposal, a crucial step is to select the objects that will take part in the study. In this case, we have selected objects whose lexical labels are available in the AlexNet pre-trained model. In other words, the class of each object must be present in the categories of the ImageNet 2012 dataset.

A preliminary study was developed with the intention of selecting the objects for the experiments. First, a series of images was captured from many vantage points of an object instance. Second, these images were processed by the labeling tool to produce a semantic descriptor of each image. In this case, each descriptor contained 1,000 labels together with their associated confidence values. Next, the images belonging to the same instance were grouped to calculate the mean confidence for each label in the semantic descriptors of the group of images. This generated a semantic descriptor for the image group. These mean values were then represented

in a cloud label as in Figure 6.3. Here, the bigger the font size in the cloud, the higher is the mean confidence assigned by the labeling tool.

Figure 6.3 left shows two objects whose grouped instances obtained a high assertion rate, as represented in the cloud label. The largest label corresponds to the real class of the group, in this case *ocarina* and *hair spray*, whereas Figure 6.3 right corresponds to a pair of objects with a low assertion rate, due to their grouped instances not corresponding to their real class. In this case *joystick* was misclassified as *hair slide* as well as *tennis ball* as *knee pad*, as presented in the label cloud.



**Figure 6.3:** Cloud tags representing the labeling tool high (left) and low (right) assertion rate for object instances selected for the study.

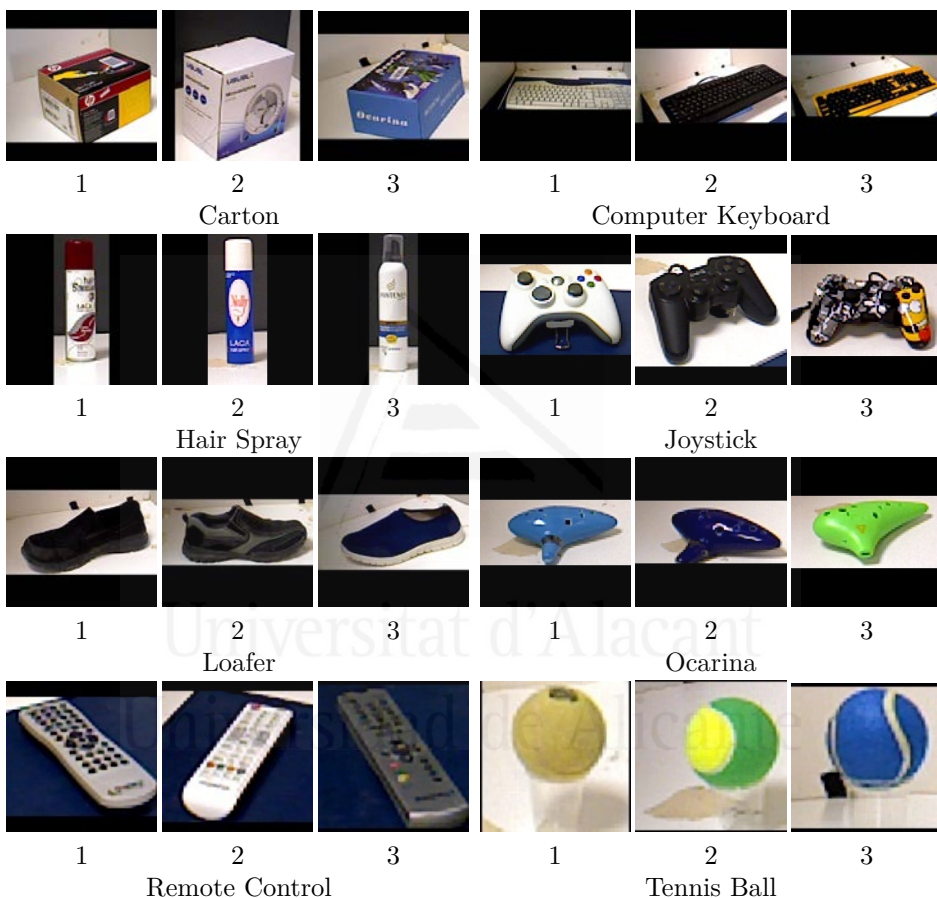
In order to select the objects we captured 3 instances of every object. Then, analyzing the cloud tags, we identified the objects obtaining the lowest assertion rate. These objects have a high geometric similarity with the rest of the instances, but with several visual differences such as colors or textures as shown in Figure 6.3 right.

Hence, for each selected object there are three different instances. For instance, for the object *ocarina* there are three different real ocarinas for use in the study.

After the preliminary study, we selected a total of eight objects for the experimentation, namely: *carton*, *computer keyboard*, *hair spray*, *joystick*,

*loafer, ocarina, remote control and tennis ball.*

Figure 6.4 shows the objects selected for the experiments in this chapter. The selection gives us a total of eight objects, with three instances for each one, generating a total of 24 different instances in the whole experiment.



**Figure 6.4:** Images used for the experiment

### 6.4.2 Experimental Setup

In order to carry out the evaluation of the proposal, we have to follow the pipeline described in previous sections, with the aim of finding the clusters for the training phase. For developing the experiments, we need

to define the labeling tool that will be used for the 2D Labeling Phase, the global 3D descriptor for the 3D Description Phase, and the Classifier algorithm for the Training and Validation Phase. These configurations are detailed in Table 6.2.

**Table 6.2:** Experimental Setup

Phase	Parameter	Selected Option
Acquisition	Object Number	8 Objects x 3 Instances
Acquisition	Samples Number	500 Samples x 8 Objects
2D Labeling	Labeling Tool	AlexNet-ImageNet Pre-Trained Caffe Model <sup>a</sup>
3D Description	3D Global Descriptor	Ensemble of Shape Functions (ESF) [Wohlkinger and Vincze, 2011]
Training and Validation	Classifier Algorithm	Support Vector Machines (SVM) [Chang and Lin, 2011]
Training and Validation	Experiment Combinations	30 Random Combinations

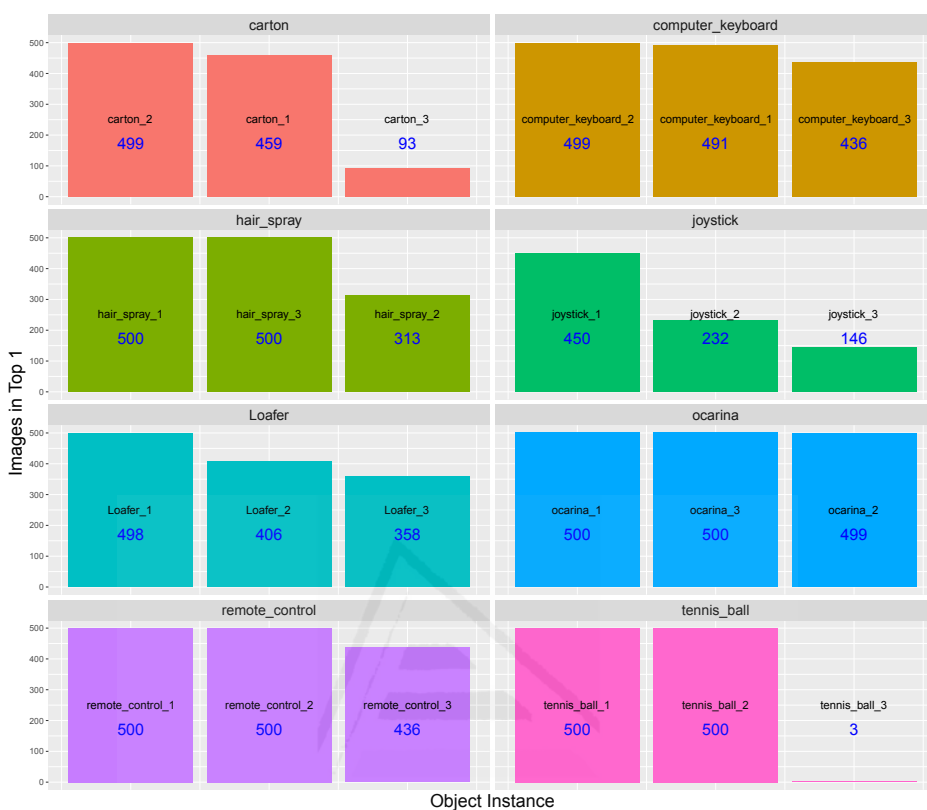
<sup>a</sup> Available at: [http://caffe.berkeleyvision.org/model\\_zoo.html](http://caffe.berkeleyvision.org/model_zoo.html)

In the 2D Labeling Phase, the lexical labels available for every image are those of the eight labels that represent the selected objects described in Section 6.4.1 and shown in Figure 6.4. Since ImageNet-AlexNet model has a total of 1,000 labels, the number of categories of the model was reduced to eight in order to fit the number of objects in the study. This phase is repeated for every cluster detected in the original point cloud.

As an outcome of this phase, Figure 6.5 shows the Top-1 result for the labeling phase of each object instance. Here, the blue numbers in the bars indicate the amount of successfully labeled images. These results will be used to construct the set of instances for the experiments.

In the 3D Description Phase of our experiments, we used the Ensemble of Shape Functions (ESF)[Wohlkinger and Vincze, 2011] to describe the previously detected segmented point clusters. The ESF descriptor have been already introduced in the Section 3.4.1.3.

Once the description phase was completed, the training phase was carried out using the Support Vector Machine Library (libsvm)[Chang and Lin, 2011] to train the classification models (SVMs) used for the different experiments. These experiments were developed using a combination



**Figure 6.5:** Top 1 labeling generated by the labeling tool.

of random object instances for the training and test sets. Experimental results will be detailed in the following sections.

### 6.4.3 Baseline Results: proposal validation

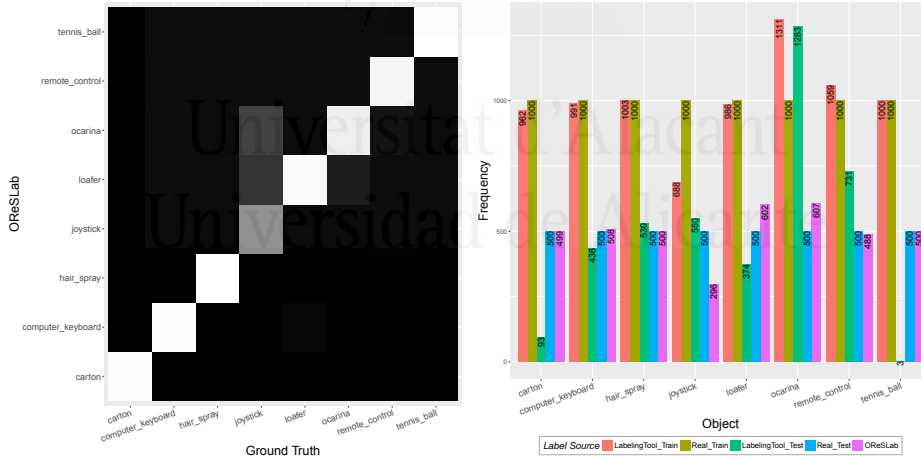
For all the experiments, we defined that a *Success* occurs when the classifier assigns the right category to the object, evaluated. If not, a *Fail* occurs. Hence, the Top-1 results generated by the labeling tool are defined as the number of instances correctly classified by the labeling tool when compared with the ground-truth category of the object.

The baseline scenario considers a training/test split where the instances with the best classification rates using the labeling tool are selected for training and the rest for testing. The objective of this scenario is to validate the proposal in a proper experiment, and then move to an experimentation

using fair scenarios.

Our baseline scenario is thus composed of the two best Top-1 object instances for training and the worst Top-1 object instance for testing, as shown in Figure 6.5. This baseline represents a worst-case scenario for a 2D classifier, where it cannot adequately generalize due to the visual differences in instances of a same object. This train/test split, then, is designed to outline the robustness that a 3D-based classifier can provide for a vision system deployed in a real scenario.

Using this set, the accuracy obtained by the classification model is 93.5%, whereas the accuracy obtained for the same instance combination with the labeling tool was 57.1%. Hence, the baseline configuration outperforms the results produced by the labeling tool when the baseline set is tested. The confusion matrix for the baseline test is shown in Figure 6.6 left. This experiment was carried out aiming to determinate whether the proposed pipeline is able to improve the recognition results of the labeling tool in a classification test.





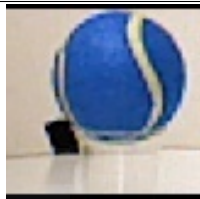
**Figure 6.6:** Confusion Matrix (left) and set distribution (right) for the baseline set experiment.

Figure 6.6 right shows the distribution of the categories in the training sequence. We can appreciate the misclassification produced by the labeling tool for a number of objects in the case study.


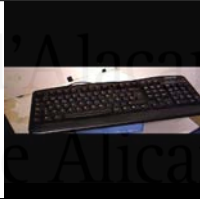
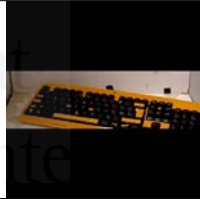
Tables 6.3 and 6.4 present a Success and a Fail case, respectively, for

the baseline experiment set. In Table 6.3, due to the behavior of the CNN, the blue *tennis ball* was labeled by the labeling tool as an *ocarina*, which is based on the visual appearance of the image. The fail case presented in Table 6.4, where a *computer keyboard* is misclassified as a *remote control* is also striking.

**Table 6.3:** Success case from the baseline experiment

	Training Instances		Test Instance
			
<b>Ground Truth</b>	tennis ball	tennis ball	tennis ball
<b>Labeling Tool Class</b>	tennis ball	tennis ball	ocarina
<b>OResLab Class</b>			tennis ball

**Table 6.4:** Fail case from the baseline experiment

	Training Instances		Test Instance
			
<b>Ground Truth</b>	comp. keyboard	comp. keyboard	comp. keyboard
<b>Labeling Tool Class</b>	comp. keyboard	comp. keyboard	comp. keyboard
<b>OResLab Class</b>			remote control

#### 6.4.4 Experimental Results

Figure 6.7 presents the accuracy values obtained for 30 randomly created experiment combinations. This combination was made in order to produce a more realistic scenario for testing the robustness of the system, even in combinations where the training instances are not those best available.

For each experiment, the training set was composed of two object in-



stances, whereas the test set had the remaining instance for the object. The left side of this chart represents the accuracy obtained by the labeling tool (a Caffe model pre-trained for our experiments) and the right side, the accuracy produced by the classification model. The green labels indicate the experiment where the trained classification model outperforms the labeling tool as classifier. The experiment set obtained a mean accuracy of 84.55%.

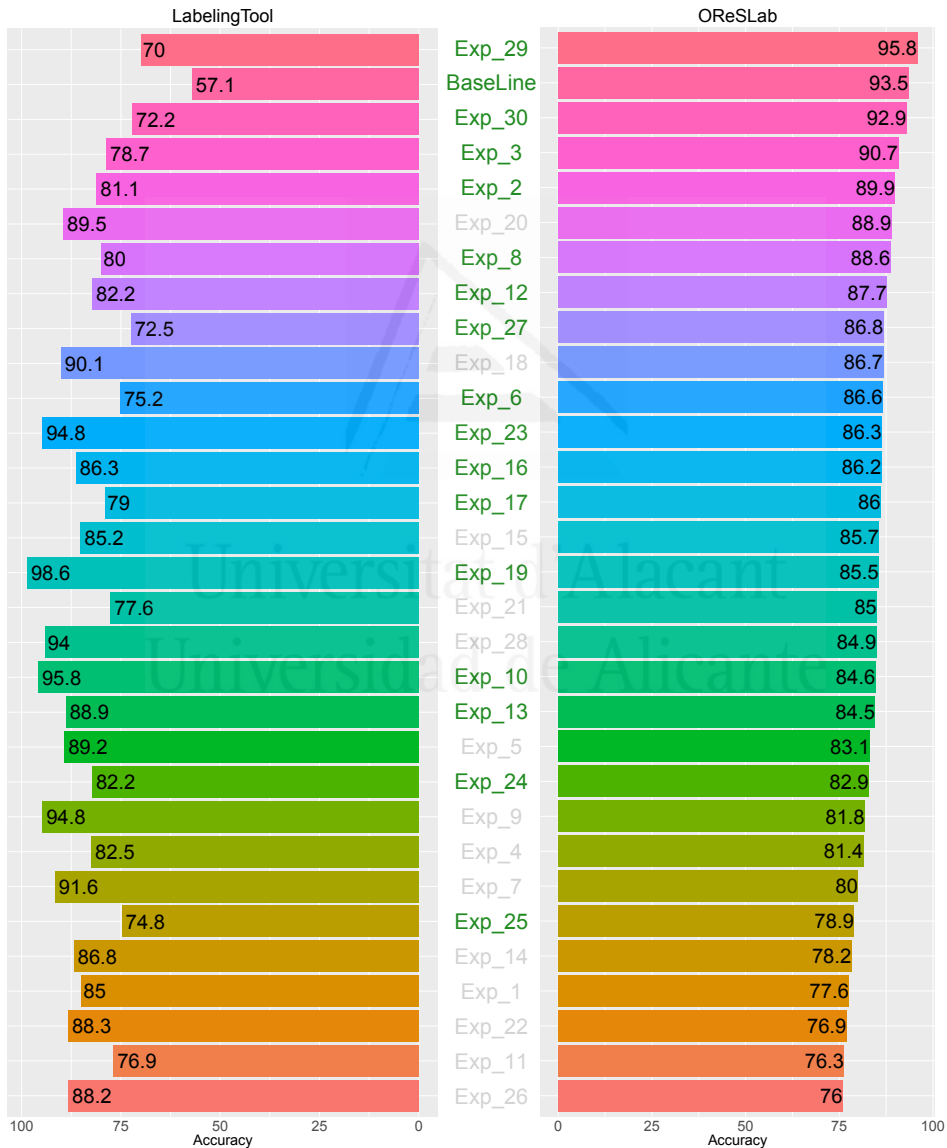





Figure 6.7: Accuracy obtained by every combination.

Tables 6.5 and 6.6 present a success and a fail case, respectively, for the experiment with the lowest accuracy rate. In Table 6.5, a loafer is accurately classified by the classification model, although one training instance is erroneously classified by the labeling tool. On the other hand, Table 6.6 shows a scenario where the *carton* was not recognized either by the labeling tool or the classification model.

**Table 6.5:** Success case from the lowest accuracy experiment

	Training Instances		Test Instance
			
<b>Ground Truth</b>	loafer	loafer	loafer
<b>Labeling Tool Class</b>	ocarina	loafer	loafer
<b>OReSLab Class</b>			loafer

**Table 6.6:** Fail case from the lowest accuracy experiment




	Training Instances		Test Instance
			
<b>Ground Truth</b>	carton	carton	carton
<b>Labeling Tool Class</b>	remote control	carton	remote control
<b>OReSLab Class</b>			comp. keyboard

Table 6.7 presents the composition of the training and test set for the experiments with the highest and lowest accuracy, as well as the baseline combination.

Table 6.8 shows the object instances which have the lowest Top-1 results for the labeling tool during the labeling phase, and their apparition in the training set of the lowest accuracy experiments.

**Table 6.7:** Training Instances Comparison

<b>Lowest Accuracy Combination</b>	<b>Highest Accuracy Combination</b>	<b>Baseline Combination</b>
Exp_26 Acc=76.0%	Exp_29 Acc=95.8%	Baseline Acc=93.5%
<b>Training Instances</b>		
carton_2	carton_2	carton_1
carton_3	carton_3	carton_2
computer_keyboard_2	computer_keyboard_1	computer_keyboard_1
computer_keyboard_3	computer_keyboard_2	computer_keyboard_2
hair_spray_1	hair_spray_1	hair_spray_1
hair_spray_2	hair_spray_3	hair_spray_3
joystick_1	joystick_1	joystick_1
joystick_2	joystick_3	joystick_2
Loafer_2	Loafer_1	Loafer_1
Loafer_3	Loafer_2	Loafer_2
ocarina_1	ocarina_1	ocarina_1
ocarina_2	ocarina_3	ocarina_3
remote_control_1	remote_control_3	remote_control_1
remote_control_2	remote_control_2	remote_control_2
tennis_ball_1	tennis_ball_1	tennis_ball_1
tennis_ball_3	tennis_ball_2	tennis_ball_2
<b>Test Instances</b>		
carton_1	carton_1	carton_3
computer_keyboard_1	computer_keyboard_3	computer_keyboard_3
hair_spray_3	hair_spray_2	hair_spray_2
joystick_3	joystick_2	joystick_3
Loafer_1	Loafer_3	Loafer_3
ocarina_3	ocarina_2	ocarina_2
remote_control_3	remote_control_1	remote_control_3
tennis_ball_2	tennis_ball_3	tennis_ball_3

## 6.5 Discussion

The aim of the research presented in this chapter was to find a novel way to recognize objects based on the use of 3D information. This type of recognition is currently confronted by several obstacles such as dark environments, occlusions, as well as the lack of sufficient object datasets for training. Therefore, this chapter proposes the use of a 2D labeling

**Table 6.8:** Common instances in the training set of the lowest accuracy experiments

<b>Object Instance</b>	<b>Experiment</b>				
	<b>14</b>	<b>1</b>	<b>22</b>	<b>11</b>	<b>26</b>
<b>carton_3</b>	✓			✓	✓
<b>joystick_3</b>		✓	✓	✓	
<b>tennis_ball_3</b>	✓	✓	✓		✓
<b>Accuracy(%)</b>	<b>78.2</b>	<b>77.6</b>	<b>76.9</b>	<b>76.3</b>	<b>76.0</b>

system for assigning categories to objects and describing them with a 3D feature descriptor in order to train a classification model. Our results support the initial premise and demonstrate the suitability of the solution to the problems described.

The experiments developed for the validation of the proposal show that the procedure developed has the capacity to recognize unseen objects based only on their 3D features, without the need to train a classifier using a highly similar instance of the object. Our results also demonstrate that the use of 3D data help to overcome certain difficulties faced by 2D classification systems, such as the difference in visual appearance between objects in the training and test sets.

The CNN model has shown its capacity to accurately classify the projections of the points in a point cloud. Nevertheless, the use of this DL-based labeling systems creates errors in the datasets used for training the classification models. However, these errors do not significantly affect the accuracy of the method when a reduced number of misclassified instances are used for training the model. Hence, the use of the 3D features enables us to build a robust object classifier, due to the working method of the classification algorithm that groups similar features together.

The findings in this research facilitate the adaptation of the proposed method for using in challenging areas with several different lighting conditions. Furthermore, the generalization capability of the classifier allows a small degree of independence for the systems, robot or platform that select the proposed pipeline for classification issues.

## 6.6 Conclusions and future work

In this chapter, we have proposed the use of an external labeling tool for assigning a lexical label to a cluster of points detected in a point cloud. Then, with these clusters, a classifier is trained to recognize instances of the clusters that are difficult to classify using a 2D classification method. The labeling tool could be any kind of system with the capacity of produce labels and a confidence value for an image.

Experimental results show that even though some instances were erroneously classified by the labeling tool, our classification model is able to recognize the objects based on their 3D features. Results also show the advantages of using combined 2D and 3D procedures in order to make the most of the generalization capabilities of 2D classification models as well as the morphology information provided by the 3D data.

As future work, we plan to integrate the classification model in a mobile robot, to detect objects in an environment or location, and make it possible to grasp or manipulate them.

# Conclusions

---

This chapter discusses the main conclusions extracted from the work presented in this dissertation. The chapter is organized in four sections: Section 7.1 presents and discusses the final conclusions of the work presented in this thesis. Section 7.2 enumerates the most relevant contributions made in the topic of research. Next, Section 7.3 lists the publications derived from the presented work. Finally, Section 7.4 presents future works: open problems and research topics that remain for future research.

## 7.1 Conclusions

In this thesis we focus on solving scene understanding problems for mobile robots. These kinds of problems are usually addressed by means of object recognition procedures. Hence, we first presented a study for testing how to manage noise in 3D clouds in order to accurately identify objects. In this study, Growing Neural Gas (GNG) was used to represent noisy clouds which were then compared with other 3D representation methods. However, although GNG produced positive results, 3D object recognition still had to deal with several problems as well as computational requirements. Consequently, we decided to explore the possibility of using 2D data in the procedure.

Furthermore, after studying 2D recognition approaches, we came up with a novel method to represent images, based on semantic labels pro-

duced with a Convolutional Neural Network (CNN) model. These semantic labels also include a confidence value representing how likely it is for the label to be present in the image. These pairs of values (label-confidence) will be used as a descriptor for images. The proposed method employs deep learning techniques, but it does not need long training periods in order to build the classification/labeling models. Therefore, the proposed method could be adapted to any labeling systems that produce a label with a corresponding confidence value.

The semantic descriptor allows us to work with images and obtain measures for image comparison in order to group them by similarity. Using this descriptor, we could infer semantic information from an image or a group of images. This semantic information could describe the composition of the images (object presence) as well as their probable semantic category (room, kitchen). Therefore, semantic descriptors contribute not only as a regular image representation, but also with their semantic description capability.

In addition, we evaluated the semantic descriptor in supervised indoors scene classification problems. Results demonstrate the goodness of the semantic descriptor when compared with traditional numeric descriptors.

Moreover, the semantic descriptor was tested in Topological and Semantic Mapping issues. Here, the use of an accurate image representation makes the constructed maps robust. In mapping situations the semantic descriptor supplies a wide description of possible objects present in the room. In addition, the category of places for semantic mapping could be derived only by taking into account the labels with highest mean confidence, obtained by grouping similar images in a sequence. Similarly, this inference procedure could be applied to topological maps where the groups correspond to nodes of similar images in a sequence.

Last but not least, CNN models have been tested as a labeling system for clusters of points segmented from point clouds, in order to train a classifier that could accurately recognize 3D objects that would normally be difficult to detect when using 2D image classifiers.

Finally, the studies carried out in this thesis allow us to confirm that using semantic descriptors produced by some labeling external system (in

our case, a CNN model) makes it possible to achieve detailed understanding of the scenes as well as their semantic meaning.

## 7.2 Contributions of the Thesis

The contributions made in this body of research are as follows:

- The application of the GNG-based algorithm to represent noisy scene and develop object recognition in these scenes.
- The use of pre-trained CNN models as a labeling system to build a semantic image descriptor.
- The application of a semantic descriptor in mapping procedures in order to infer semantic information about places.
- The training of 3D objects classifiers with objects labeled using a 2D pre-trained CNN model.

## 7.3 Publications

The following articles were published as a result of the research carried out by the doctoral candidate:

- Published articles in scientific journals:
  - José Carlos Rangel, Jesús Martínez-Gómez, Cristina Romero-González, Ismael García-Varea and Miguel Cazorla: OReSLab: 3D Object Recognition through CNN Semi-Supervised Labeling. Under Review. **Applied Soft Computing** Impact Factor (JCR 2015): 2.857.
  - José Carlos Rangel, Miguel Cazorla, Ismael García-Varea, Cristina Romero-González and Jesús Martínez-Gómez: AuSeMap: Automatic Semantic Map Generation. Under Review. **Autonomous Robots** Impact Factor (JCR 2015): 1.547.



- José Carlos Rangel, Jesús Martínez-Gómez, Ismael García-Varea and Miguel Cazorla: LexToMap: Lexical-based Topological Mapping. **Advanced Robotics** 31(5):268-281 (2017) Impact Factor (JCR 2015): 0.516.
- José Carlos Rangel, Miguel Cazorla, Ismael García-Varea, Jesús Martínez-Gómez, Elisa Fromont and Marc Sebban: Scene Classification based on Semantic Labeling. **Advanced Robotics** 30(11-12):758-769 (2016) Impact Factor (JCR 2015): 0.516.
- José Carlos Rangel, Vicente Morell, Miguel Cazorla, Sergio Orts-Escolano and José García-Rodríguez: Object recognition in noisy RGB-D data using GNG. **Pattern Analysis and Applications** Accepted (2016) Impact Factor (JCR 2015): 1.104.

- International conferences:
  - José Carlos Rangel , Miguel Cazorla, Ismael García-Varea, Jesús Martínez-Gómez, Élica Fromont and Marc Sebban: **Computing Image Descriptors from Annotations Acquired from External Tools**. Robot 2015: Second Iberian Robotics Conference, Lisbon, Portugal: 673-683, November 19-21.
  - José Carlos Rangel, Vicente Morell, Miguel Cazorla, Sergio Orts-Escolano and José García-Rodríguez: **Using gng on 3d object recognition in noisy rgb-d data**. International Joint Conference on Neural Networks, IJCNN 2015, Killarney, Ireland, July 12-17.
  - José Carlos Rangel, Vicente Morell, Miguel Cazorla, Sergio Orts-Escolano and José García-Rodríguez: **Object Recognition in Noisy RGB-D Data**. International work-conference on the interplay between natural and artificial computation, IWINAC 2015, Elche, Spain: 261-270, June 1-5.

## 7.4 Future Work

As a future work we propose the integration of the developed semantic classification methods in a mobile robotic platform in order to achieve a mobile platform able to analyze places in real time, which could be used in home assistance.

Regarding the semantic descriptor, we plan to continue studying its descriptive capabilities by selecting labels with high probability values and generating a reduced version of the descriptor. This procedure will take into account the actual category of the image for label selection.

Moreover, we propose the use of several models in conjunction; in other words, the application of several models to the same set of images. Each model would be focus on different aspects, such as scenes, objects, instru-

ments, furniture, etc., with the objective of obtaining the semantic classification of a place as well as its possible content. This approach could strengthen the scene understanding of a specific place.

We also propose the study of the faculties that the semantic descriptor has, in order to use them in outdoor scenes, where they could face challenges such as a greater perspective of the environment, greater number of elements in the scenes, different types and sources of illumination, as well as elements whose size is imperceptible for the neural network.

For the generation of semantic descriptors we propose the use of novel designs of CNN architectures, such as ResNet and other pre-trained models and frameworks such as Keras with TensorFlow, to complete a comparative study of these models with the ones used in this thesis and achieve an evaluation metrics that will eventually lead to the selection of the model that best fits each kind of problem.

# Appendices



Universitat d'Alacant  
Universidad de Alicante



# Resumen

---

En este apéndice se introducirá la motivación y objetivos para la realización de esta tesis. Se describirá el estado del arte relacionado con la comprensión de escenas, los conjuntos de datos utilizados durante la experimentación y las técnicas empleadas para los experimentos. Por último, se presentarán las conclusiones derivadas de los estudios y las propuestas de trabajos futuros.

## A.1 Introducción

En esta tesis doctoral se han llevado a cabo investigaciones teóricas y prácticas donde se abordó el problema de encontrar un modelo preciso y adecuado para mejorar la comprensión de escenas. Proponemos enfocarnos en los métodos de representación de escenas que mejor describan su contenido con el objetivo de realizar una clasificación semántica de grupos de imágenes. Realizamos estudios con novedosos métodos para la descripción de escenas basados en etiquetas semánticas, utilizando técnicas de *Deep Learning* aplicadas a datos capturados del mundo real. Además, los métodos de representación seleccionados serán probados en problemas de mapeado semántico y topológico, campos en los cuales los métodos tradicionales todavía afrontan problemas por ser resueltos.

Esta tesis ha sido desarrollada bajo el marco de los siguientes proyectos:

- *RETOGAR: Retorno al hogar. Sistema de mejora de la autonomía de personas con daño cerebral adquirido y dependientes en su inte-*

*gración en la sociedad.* Financiado por el Ministerio de Economía y Competitividad de España, soportado con fondos FEDER. DPI2016-76515-R.

- *SIRMAVED: Desarrollo de un sistema integral robótico de monitorización e interacción para personas con daño cerebral adquirido y dependientes.* Financiado por el Ministerio de Economía y Competitividad de España, soportado con fondos FEDER. DPI2013-40534-R.

También bajo la siguiente beca:

- Beca IFARHU 8-2014-166, del Programa IFARHU-UTP 2014 de Becas Doctorales ofrecido por el Gobierno de la República de Panamá.

Además, parte del trabajo presentado en esta tesis fue realizado durante mi estancia en el Grupo de Investigación *Data Intelligence* en la Universidad Jean Monnet en Saint-Etienne, Francia, la cual tuvo una duración de tres meses y fue financiada por la Escuela de Doctorado de la Universidad de Alicante. El trabajo realizado durante la estancia fue parcialmente supervisado por el profesor Marc Sebban.

## A.2 Motivación

Este documento es el resultado de los estudios doctorales llevados a cabo en el programa de Doctorado en *Informática* entre los años 2014 y 2017, en el Instituto Universitario de Investigación Informática de la Universidad de Alicante en España. Esta tesis se deriva de una beca de cuatro años otorgada a mí por el Gobierno de la República de Panamá en una colaboración de la *Universidad Tecnológica de Panamá* y el *Instituto para la Formación y Aprovechamiento de los Recursos Humanos* (IFARHU). La motivación para este proyecto de tesis surge por la participación y colaboración en diferentes proyectos relacionados con la visión por ordenador y robots móviles.

Los métodos de Localización y Mapeado Simultáneos (SLAM por sus siglas en inglés) es una técnica común en el campo de la robótica móvil y la cual ha sido ampliamente estudiada desde que fue presentada originalmente

por [Leonard and Durrant-Whyte, 1991] y basada en los recientes trabajos de [Smith et al., 1987]. Esta técnica se enfoca en la localización de un robot en un ambiente, así como también el mapeado del entorno para permitir al robot navegar el mismo utilizando los mapas creados.

Sin embargo, los métodos de SLAM no tienen la capacidad de informarle a un robot el tipo de lugar en donde está ubicado. Esta información debe ser adquirida mediante un análisis del entorno y produciendo con esta una descripción semántica del lugar. Tradicionalmente, la descripción semántica de lugares se ha realizado mediante la identificación de los objetos en las escenas. Por lo tanto, en la actualidad la descripción semántica es usualmente relacionada con el reconocimiento de objetos, así como también con la comprensión de escenas. No obstante, el reconocimiento de objetos todavía debe tratar con problemas como oclusiones, rotaciones y vistas parciales de los objetos, entre otros. Por consiguiente, esta línea de investigación requiere nuevas estrategias para trabajar con la información.

La categorización semántica de lugares es necesaria para los robots móviles. Estos robots deben ser capaces de entender órdenes humanas relacionadas con salas o lugares de un entorno, por ejemplo, una cocina en una casa. En consecuencia, los robots requieren un método para inferir su ubicación así como también inferir una locación nueva al mismo tiempo que la esté viendo. En esta tesis nos enfocamos en encontrar una representación precisa del entorno con el objetivo de obtener una descripción semántica de los lugares.

En años recientes con la llegada de mejores recursos computacionales, como las tarjetas gráficas GP-GPU, mejores procesadores y incremento de módulos de memoria disponibles, entre otros han permitido implementar algunas técnicas definidas en años anteriores, pero que demandaban una gran capacidad computacional. Entre estas técnicas podemos mencionar las Redes Neuronales Convolucionales (Convolutional Neural Networks, CNN por sus siglas en inglés). Estas RRNN tienen la capacidad de clasificar una imagen basándose en su apariencia visual. Por otro lado, las CNNs nos permiten obtener una lista de etiquetas con un valor asociado que representa la probabilidad de la presencia de un objeto en la escena. Los posibles objetos en esta lista son derivados de conjuntos de entrenamien-



to, los cuales la red ha aprendido a reconocer. Por lo tanto, esta lista de probabilidades permite describir un entorno. En esta tesis nos enfocamos en la utilización de dicha lista de probabilidades como una representación eficiente del entorno y entonces asignar una categoría semántica a las regiones donde un robot móvil tiene la capacidad de moverse utilizando un mapa que pudo haber sido construido empleando técnicas de SLAM. Otro trabajo llevado a cabo durante esta tesis ha sido la construcción de mapas topológicos y semánticos utilizando para ello los descriptores semánticos generados utilizando las CNNs.

Durante mi colaboración con el Grupo de Investigación *Data Intelligence*, trabajé en el diseño de una representación semántica de los objetos que están presentes en una escena, basado en técnicas de *deep learning*. Esta representación era requerida para el desarrollo de un sistema en tiempo real que detectase el tipo de lugar donde se encuentra un robot ubicado.

### A.3 Trabajos Relacionados

En esta tesis tratamos con problemas relacionados con la comprensión de escenas guiada por reconocimiento de objetos. Comúnmente, la comprensión de escenas ha sido dirigida mediante la identificación de los objetos que están presentes en las escenas. Como se propuso en [Li et al., 2009] donde los autores desarrollaron un sistema que usando imágenes etiquetadas de Flickr.com, tiene la capacidad de clasificar, anotar y segmentar imágenes de una variedad de escenas. Este sistema emplea un modelo jerárquico para unificar la información desde varios niveles (objeto, parche y escena).

El trabajo presentado en [Liao et al., 2016] aprovecha las capacidades de una CNN para aprender las características de los objetos con el fin de construir un modelo de CNN para clasificación de escenas con regularización de segmentación (SS-CNN). En este trabajo los autores toman ventaja de la interacción de los objetos y las escenas para ejecutar una segmentación semántica basada en información previa sobre la ocurrencia de los objetos. La información a nivel de objeto es aprendida en etapas tempranas del proceso debido a que estas están relacionadas con las características

aprendidas para la clasificación de escenas.

En las investigaciones desarrolladas por [Nguyen et al., 2016] se propone utilizar una CNN multi-nivel para extraer las características de una imagen. Estas se extraen tomando la salida de la última capa oculta de la red, en otras palabras, la capa totalmente conectada justo antes de la capa final de clasificación. Las características CNN son extraídas en dos niveles: global y de región. Primero, las características globales son empleadas para buscar imágenes similares en una base de datos de recuperación. Como siguiente paso, una vez que las imágenes similares han sido encontradas, las características de región se calculan desde super píxeles. Estas últimas incluyen también ciertas características seleccionadas ad-hoc (forma, textura, etc). Entonces utilizando una mezcla de características de región, se calculan las similitudes entre un conjunto de etiquetas, calculando un valor de probabilidad y así producir el análisis de la imagen. De forma similar el algoritmo CollageParsing [Tung and Little, 2016] emplea Campos Aleatorios de Markov (Markov Random Fields MRF) para encontrar imágenes relacionadas, pero en lugar de usar comparación a nivel de super píxeles para realizar la transferencia de etiquetas, los autores definieron una ventana adaptativa al contenido. Estas ventanas son empleadas para calcular potenciales operadores unarios, mediante el emparejamiento de contenido similar en el conjunto de recuperación de la imagen.

En lo concerniente a conjuntos de datos estándar para la comprensión de escenas, SUN RGD-B [Song et al., 2015] propone un escenario para probar enfoques de comprensión de escenas. En este las imágenes contienen anotaciones, cajas delimitadoras y polígonos. Este conjunto ha sido evaluado en varias tareas como la categorización de escenas, segmentación semántica, detección de objetos, orientación de objetos, disposición de escenas y comprensión de escenas. Por otra parte, el sistema ScanNet [Handa et al., 2016] consiste en escenas sintéticas anotadas que incluyen objetos y etiquetas para las escenas. Estas escenas contienen objetos muestreados de varios conjuntos de datos como ModelNet y texturas de OpenSurfaces y ArchiveTextures. El sistema provee a los usuarios la oportunidad de generar escenas anotadas basadas en los objetos disponibles en el conjunto de datos. La base de datos de características SUN [Patterson et al., 2014] está

compuesta por un conjunto de imágenes anotadas (por humanos) con sus características y pertenecientes a 707 categorías diferentes. Este trabajo estudia la interrelación entre categorías y características, prueba el conjunto de datos en tareas como clasificación de escenas, aprendizaje *zero-shot* y búsqueda semántica de imágenes entre otras. El conjunto de datos *Places* [Zhou et al., 2014] es un conjunto de imágenes centrado en escenas que está compuesto por una gran cantidad de imágenes etiquetadas con su categoría. Este conjunto de datos ha sido probado en tareas de clasificación y mediante el empleo de *deep learning* para el desarrollo de modelos de clasificación.

La comprensión de escenas ha sido abordada desde varios enfoques como con los Modelos Jerárquicos Bayesianos en [Steinberg et al., 2015] los cuales se enfocan en utilizar técnicas no supervisadas o solo basadas en datos visuales. Estas técnicas se pueden aplicar para obtener una comprensión adecuada de la escena. El trabajo propuesto por [Li et al., 2015] planea utilizar un *hashing* sensitivo local (LSH por sus siglas en inglés) para superar el problema de la polisemia visual y el concepto de polimorfismo (VPCP por sus siglas en inglés) para mejorar la comprensión de escenas en imágenes de gran tamaño. La propuesta de [Redondo-Cabrera et al., 2014] introduce un modelo de *Bag-Of-Words* 3D para la categorización de objetos y escenas. El modelo emplea descriptores 3D cuantificados obtenidos de nubes de puntos.

La propuesta de [Vaca-Castano et al., 2017] estudia cómo los objetos están relacionados a una escena en restricciones de visión egocéntricas. Este trabajo propone mejorar la detección de objetos primero mediante la predicción de la categoría de la escena y después de acuerdo a esa categoría, computar los objetos que se pueden encontrar presentes en la escena. Los autores incluso presentan un sistema en el cual utilizando redes de memoria a corto plazo (*Long Short-Term Memory Networks*, LSTM por sus siglas en inglés), la detección preliminar de la categoría de la escena no es necesaria.

Los autores en [Tung and Little, 2015] emplean una versión concatenada de detectores de objetos producida por un *Never Ending Image Learning* (NEIL) y modelos de ImageNet para aprender los objetos y sus relaciones en las escenas. Esta concatenación fue desarrollada para mejorar

el reconocimiento de características en las imágenes.

## A.4 Conjuntos de Datos

Esta sección introduce al lector a los conjuntos de datos que serán utilizados como conjuntos de validación para los métodos propuestos. Varios conjuntos de datos públicos se han empleado para validar nuestras propuestas.

### A.4.1 SHOT

Con el objetivo de experimentar los enfoques de reconocimiento de objetos, como primer paso en la comprensión de escenas, hemos experimentado con métodos de reconocimiento de objetos 3D. Empleamos el conjunto de datos SHOT<sup>1</sup> (Universidad de Bologna) [Tombari et al., 2010][Tombari and Salti, 2011][Computer Vision LAB, 2013]. Este ha sido adquirido usando una técnica estéreo (STS por sus siglas en inglés) y está compuesto por siete modelos, con diferentes puntos de vista para cada uno y 17 escenas, creando un total de 49 instancias de reconocimiento de objetos. En la Figura A.1 se muestran algunos objetos del conjunto, mientras que en la Figura A.2 se exponen algunas escenas donde aparecen diferentes objetos. Cada escena del conjunto de datos contiene un número aleatorio de objetos presentes en ella. Estos objetos pueden aparecer desde varias posiciones y puntos de vista.

### A.4.2 *Dataset* KTH:IDOL

Para la evaluación de la propuesta de mapeado topológico hemos optado por el conjunto de datos KTH-IDOL 2 [Luo et al., 2007]. Denominado Base de datos de imágenes para localización de robots (IDOL por sus siglas en inglés) es un conjunto de imágenes que proveen secuencias de imágenes en perspectiva adquiridas en tres diferentes condiciones de iluminación: soleado, nublado y nocturno. Estas secuencias fueron generadas utilizando

---

<sup>1</sup><http://www.vision.deis.unibo.it/research/80-shot>



**Figura A.1:** Ejemplo de objetos en el conjunto de datos SHOT.

dos plataformas robóticas diferentes llamadas Minnie o PeopleBot y Dumbo o PowerBot, controladas por un operador humano. Los datos verdaderos en el conjunto incluye la categoría semántica de la habitación donde se adquirió la imagen, la marca de tiempo y la pose del robot  $\langle x, y, \theta \rangle$  durante la adquisición. Cuenta con cinco categorías diferentes: pasillo (CR), cocina (KT), oficina de una persona (1-PO), oficina de dos personas (2-PO) y zona de impresora (PA). El conjunto de datos incluye cuatro diferentes secuencias para cada combinación de robot y condición de iluminación. De entre todas las secuencias, seleccionamos las 12 secuencias capturadas con el PeopleBot, en las cuales la posición de la cámara (aproximadamente un metro sobre el suelo) es la más similar a la de las plataformas de robótica actuales.

La cantidad de imágenes en el conjunto de datos por condición de iluminación y categoría semántica así como también el mapa del entorno donde se adquirieron las imágenes se muestra en la Figura A.3. La distribución de las imágenes está claramente desbalanceada debido a que la mayoría de las imágenes corresponden a la categoría de corredor. Las secuencias



Figura A.2: Ejemplo de escenas en el conjunto de datos SHOT.

Ilum.	Nublado	Noche	Soleado
Secuencias	1,2,3,4	1,2,3,4	1,2,3,4
#Img CR	1,632	1,704	1,582
#Img 1-PO	458	522	468
#Img 2-PO	518	633	518
#Img KT	656	611	556
#Img PA	488	534	482
#Img	3,752	4,004	3,606

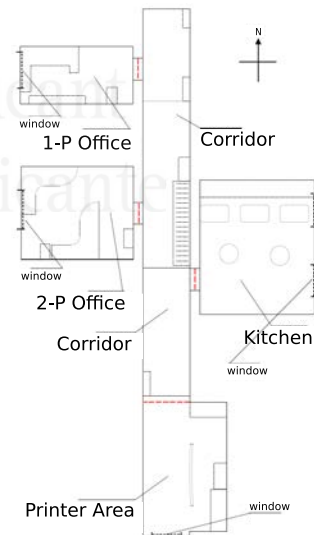


Figura A.3: Información del KTH-IDOL 2: Distribución de imágenes (izquierda) y entorno (derecha).

3-4 fueron adquiridas seis meses después que las secuencias 1-2, lo cual introduce pequeñas variaciones ambientales debido a la actividad humana. En la Figura A.4 se muestran 15 imágenes de ejemplo de este conjunto de datos. De estos ejemplos se puede observar cómo la representación visual es afectada por las condiciones de iluminación. Además, la Figura A.5 ejemplifica los efectos de la actividad humana en el entorno.



**Figura A.4:** Ejemplos de imágenes del conjunto de datos KTH-IDOL 2 adquiridas bajo tres diferentes condiciones de iluminación (filas) y diferentes categorías (columnas).

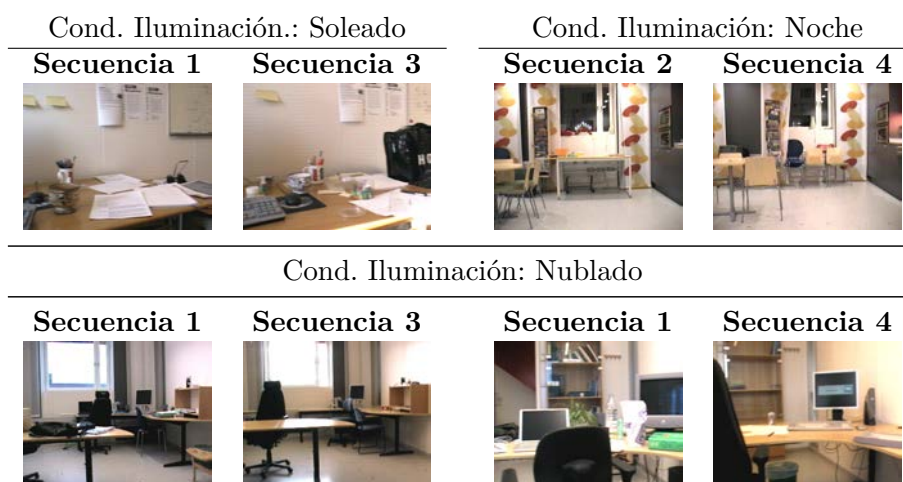
### A.4.3 ViDRILO

El conjunto de datos para localización de robots con información visual y de profundidad e información de objetos (*Visual and Depth Robot Indoor Localization with Objects information*, ViDRILO por sus siglas en inglés) ha sido creado el reto de Robot Visión de ImageClef<sup>2</sup>. Este reto va dirigido a problemas de clasificación semántica de lugares mediante información de profundidad y visual, incluyendo el reconocimiento de objetos. ViDRILO fue escogido como fuente de información para las propuestas de clasificación de escenas y mapeado semántico.

Las características principales de este conjunto<sup>3</sup> el cual provee cinco diferentes secuencias de imágenes RGB-D capturadas por un robot móvil

<sup>2</sup><http://imageclef.org>

<sup>3</sup>Disponible para descarga en <http://www.rovit.ua.es/dataset/vidrilo/>



**Figura A.5:** Imágenes ilustrando los cambios que se produjeron debido a la interacción humana en el entorno.

en un entorno interior de oficinas, se muestran en la Tabla A.1. ViDRILO fue adquirido con un espacio de tiempo de 12 meses con el objetivo de incorporar variaciones producidas por la actividad humana con el paso del tiempo.

El conjunto de datos completo fue adquirido en dos edificios de oficinas (Politécnica II y III) en la Universidad de Alicante (España) utilizando un robot PowerBot con una cámara kinect encima de este. El robot posee una unidad basculante con un láser 2D Sick. La cámara kinect fue colocada en la parte superior de la unidad basculante lo cual provee una altura total de 90cm. El robot fue tele-operado utilizando un *joystick* durante toda la ruta, desplazándose a una velocidad aproximada de 0.3m/s. El conjunto contiene imágenes RGB (imagen de color) y las nubes de puntos 3D coloreadas, para todas las secuencias tomadas en diferentes estancias y con diferentes distribuciones de objetos. La Figura A.6 muestra un ejemplo de la estancia con la categoría secretaria.

Cada imagen RGB-D está etiquetada con la categoría semántica de la escena en la cual se adquirió, de un conjunto de 10 categorías de estancias. Varias secuencias de ViDRILO fueron utilizadas como *benchmark* del reto RobotVision en las ediciones más recientes de la competición ImageCLEF [Martínez-Gómez et al., 2015]. Hemos optado por este conjunto de





**Figura A.6:** Ejemplo de una imagen del conjunto de datos ViDRILO. Imagen de color a la izquierda y nube de puntos a la derecha.

**Cuadro A.1:** Distribución de las Secuencias en ViDRILO.

Secuencia	#Imágenes	Pisos	Salas Oscuras	Espacio de Tiempo	Edificio
Secuencia 1	2,389	1,2	0/18	0 meses	A
Secuencia 2	4,579	1,2	0/18	0 meses	A
Secuencia 3	2,248	2	4/13	3 meses	A
Secuencia 4	4,826	1,2	6/18	6 meses	A
Secuencia 5	8,412	1,2	0/20	12 meses	B

datos debido a : (a) este provee secuencias de imágenes RGB-D adquiridas con una continuidad temporal; (b) las escenas están etiquetadas semánticamente; y (c) las áreas separadas espacialmente pueden compartir la misma categoría semántica, lo cual permite la generalización. La Figura A.7 muestra imágenes de las 10 categorías de ViDRILO.

## A.5 *Deep Learning* (DL)

La creciente aplicación del *Deep Learning* (DL) en la comunidad robótica ha abierto nuevas líneas de investigación en los últimos años. Además, de la generación de modelos para la resolución de problemas [Bo et al., 2013, Neverova et al., 2014], la liberación de modelos pre-entrenados permite una aplicación directa de los sistemas de DL generados [Rangel et al., 2016a]. Lo anterior se hace posible gracias a la existencia de sistemas modulares para DL como Caffe [Jia et al., 2014]. La aplicación directa



**Figura A.7:** Ejemplo de las imágenes visuales de las 10 categorías en ViDRILO.

de estos modelos pre-entrenados evita los requerimientos computacionales para realizar el proceso de aprendizaje de estos, recursos como son largos periodos de tiempo para llevar a cabo el aprendizaje/entrenamiento incluso utilizando procesamiento en GPUs y las grandes cantidades de espacio de almacenamiento para los datos de entrenamiento. De los modelos de DL existentes, debemos resaltar aquellos que se han generado de imágenes con una categoría asignada con etiquetas léxicas generalistas y heterogéneas [Krizhevsky et al., 2012, Zhou et al., 2014]. El empleo de estos modelos permite a cualquier sistema de visión por computadora etiquetar imágenes con un conjunto de etiquetas léxicas las cuales describen su contenido, lo cual ha sido recientemente demostrado in [Carneiro et al., 2015, Murthy et al., 2015, Rangel et al., 2016a].

La utilización de DL es considerado un hito destacable en áreas como visión por computador y robótica [LeCun et al., 2010]. DL es capaz de proveer clasificadores con la capacidad no solo de clasificar datos sino también de extraer características intermedios automáticamente. Esta técnica ha sido aplicada al etiquetado de imágenes con sorprendentes resultados. Por ejemplo, el equipo desarrollador de Clarifai obtuvo el primer lugar en la competencia ImageNet [Russakovsky et al., 2015] edición 2013, mediante el uso de redes neuronales convolucionales [Krizhevsky et al., 2012]. Además, de la gran cantidad de datos etiquetados/clasificados para el entrenamiento, DL requiere capacidades de procesamiento de alto nivel para conseguir el aprendizaje en sus modelos. Mientras que estos requerimien-

tos no siempre se encuentran a la vez, podemos sacar ventaja de algunas soluciones que el DL provee a través del uso de interfaces de programación de aplicaciones (API por sus siglas en inglés), como lo son Clarifai o Caffe, los cuales serán descritas más adelante.

### A.5.1 Redes Neuronales Convolucionales (CNNs)

Las CNNs son definidas como modelos de aprendizaje de máquina jerárquicos, los cuales aprenden representaciones complejas de imágenes a partir de grandes volúmenes de datos anotados [Krizhevsky et al., 2012]. Estas utilizan múltiples capas de transformaciones básicas las cuales finalmente generan una representación sofisticada de alto nivel de la imagen [Clarifai, 2015]. En la Figura A.8 se muestra la arquitectura básica de una CNN.

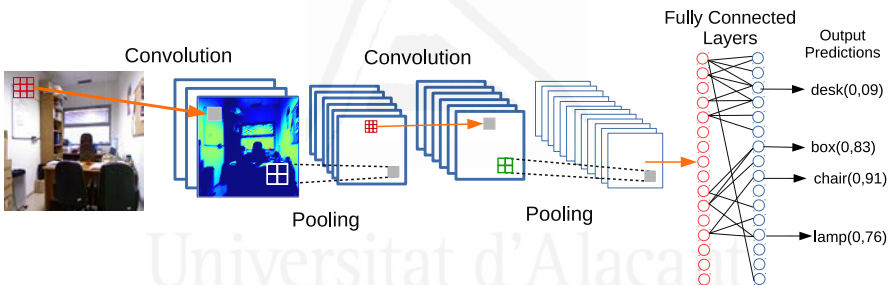


Figura A.8: Arquitectura estándar de una CNN.

Las CNN son una clase de red neuronal que se enfoca en el procesamiento de imágenes para obtener un conjunto de características, las cuales describan la imagen completamente. Esta descripción se logra mediante la concatenación de varios tipos de capas de procesamiento.

Estas redes pueden ser separadas en dos secciones diferentes las cuales podemos llamar: extracción de características y clasificación. Al inicio, la extracción de características se encarga del computo de las características de una imagen mediante la utilización de capas de convolución y *pooling*. Luego, la segunda sección, sección de clasificación, es responsable de calcular la salida de la red basándose en las características extraídas en la primera sección. Esta salida se calcula empleando capas de neuronas to-

talmente conectadas al final de la red neuronal. El tamaño (cantidad de salidas) está determinado por la cantidad de categorías en las cuales la red ha sido entrenada para reconocer.

El uso de las CNNs en problemas de clasificación requiere pasos previos para lograr producir un clasificador. La primera fase es la recolección y ordenamiento de la información (imágenes en el caso de las CNNs) la cual será utilizada para entrenar un modelo, en otras palabras, la confección del conjunto de entrenamiento. Las imágenes recolectadas se agrupan en varias categorías atendiendo a las clases que la CNN debe reconocer. Entonces, la segunda fase corresponde con el proceso de entrenamiento. Esto es un paso crucial que definirá cómo de preciso es nuestro modelo de clasificación. La fase entrenamiento de una CNN funciona mediante el análisis del conjunto completo de entrenamiento. Cada imagen del conjunto pasará a través de las capas de la CNN para enseñarle a la red a emparejar las características extraídas con la clase correspondiente.

Las capas que se encuentran entre la primera capa (capa de entrada) y la última capa (capa de salida) son denominadas capas ocultas de la red neuronal. Estas varían en número según la arquitectura de la red y las características de las imágenes a procesar.

#### A.5.1.1 Capas Convolucionales

Usualmente la primera capa de una CNN es una capa convolucional. Este tipo de capas trabaja mediante la aplicación de filtros (*kernels*) a la información de entrada. Estos filtros son aplicados a manera de ventana deslizante sobre la imagen de entrada. Consecuentemente, la imagen es parcialmente analizada mediante trozos de un tamaño definido. Los filtros, al igual que las imágenes son definidos mediante dos parámetros, altura y anchura, usualmente con igual valor. Actualmente se emplean tres canales para la información de color, pero inicialmente las imágenes de entrada fueron en escala de grises. Por consiguiente, el tamaño de un filtro se puede definir como:  $5 \times 5 \times 3$ .

Por lo tanto, cuando un filtro se aplica sobre una porción de imagen, se lleva a cabo una multiplicación elemento a elemento y sus resultados son sumados para obtener la respuesta del *kernel* para la porción corres-

pendiente de la imagen. Esta multiplicación (o aplicación de *kernel*) es repetida hasta que la imagen completa haya sido analizada. La salida del filtro produce un mapa de activación de los datos de entrada donde este mapa está compuesto del conjunto completo de respuestas del *kernel* para la imagen. Usualmente este mapa tiene un tamaño menor al de la imagen original, aunque otros enfoques se pueden utilizar para manejar los bordes de la imagen. Comúnmente los filtros incluyen un valor de desplazamiento el cual determina la cantidad de píxeles que el filtro se moverá en una dirección definida.

Una capa convolucional puede tener varios tipos de filtros, por lo cual, la salida de una capa convolucional no es solamente un mapa de activación, sino un conjunto de estos, uno por cada *kernel* en la capa convolucional. La Figura A.9 muestra el funcionamiento de un *kernel* de convolución.

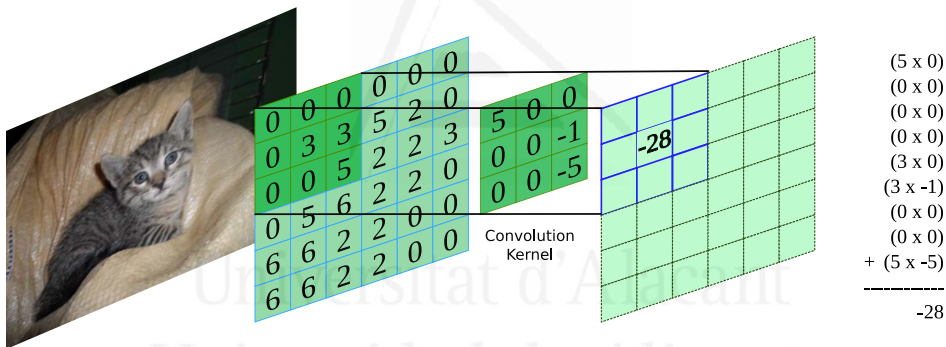


Figura A.9: Ejemplo de convolución.

Los *kernels* en las capas convolucionales, buscan la aparición de características específicas en la imagen de entrada. Estos *kernels* usualmente empiezan identificando características simples como puntos o curvas en la primera capa convolucional, luego las capas siguientes se especializan en identificar grupos de estas características y así sucesivamente encontrar características complejas y de más alto nivel como cajas o círculos.

### A.5.1.2 Funciones de Activación de las Neuronas

Las Unidades de Rectificación Lineal o ReLUs (por sus siglas en inglés de *Rectifier Linear Units*) utilizan la salida de una capa convolucional y

aplican la función de activación no saturada  $f(x) = \max(0, x)$  (Figura A.10 izquierda). La utilización de las ReLUs en procedimientos de entrenamiento mejora el mismo, reduciendo el tiempo requerido para entrenar un modelo y de la misma manera ayuda a resolver el problema del gradiente evanescente que se presenta en estas redes. El objetivo de esta función es la introducción de la no-linealidad al sistema, debido a que las capas convolucionales computan operaciones lineales (multiplicaciones elemento a elemento y sumatorios). Las funciones de activación ayudan a elevar las características no lineales del modelo y de la red. Este elevamiento se lleva a cabo sin modificar los campos respectivos de las capas convolucionales. En el pasado la modernización de la salida de una neurona estaba a cargo de funciones no lineales saturadas, especialmente la tangente hiperbólica  $f(x) = \tanh(x)$  (Figura A.10 centro), y la función sigmoidea  $f(x) = (1 + e^{-x})^{-1}$  (Figura A.10 derecha). Estas eran más lentas que las ReLUs, en términos de entrenamiento cuando se utilizaba la optimización mediante el algoritmo de gradiente descendente.

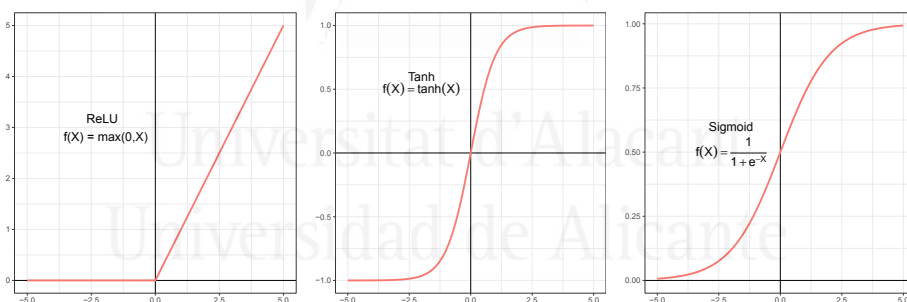


Figura A.10: Funciones de Activación estándar

### A.5.1.3 Capas de *Pooling*

Las capas de *pooling* actúan como un procedimiento de muestreo. Estas capas toman como entrada cada mapa de activación producida por una capa convolucional. De igual manera que las capas convolucionales, estas capas trabajan a manera de ventana deslizante con un tamaño definido. Por lo que, esta capa trabaja moviendo la ventana sobre la capa de activación, seleccionando un grupo de píxeles de la capa de activación y luego

utilizando el valor máximo, mínimo o la media del grupo, reemplaza el grupo por dicho valor. Por lo cual, la salida de la capa de *pooling* es de menor tamaño que el del mapa de activación.

Esta secuencia de convolución y *pooling* es el esquema básico del esquema de trabajo de una CNN, pero no todo el tiempo una capa de *pooling* sigue a una capa de convolución. Algunas arquitecturas incluyen varias capas de convolución una después de otra. Es la arquitectura de la CNN la cual define la manera en que se dará la interacción entre las capas. La Figura A.11 muestra un ejemplo de *max pooling* para uno de los canales de un mapa de activación de una capa convolucional.

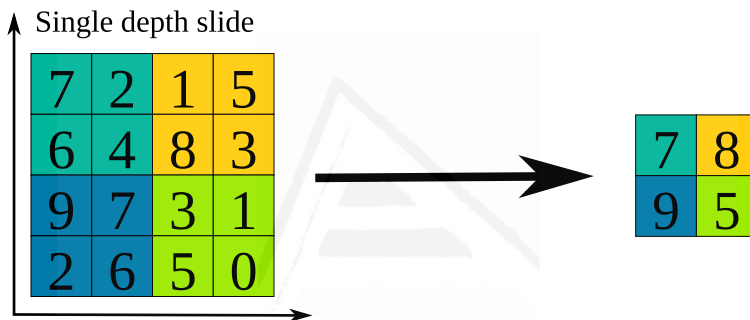


Figura A.11: *Max Pooling*.

#### A.5.1.4 Capas Totalmente Conectadas

En una CNN las últimas capas corresponden a las capas totalmente conectadas las cuales se encargan de generar la salida de la red. Este tipo de capa trabaja de manera similar a un perceptrón multicapa (Figura A.12), produciendo un vector n-dimensional como salida. El tamaño del vector dependerá de la cantidad de categorías entre la cuales la CNN es capaz de elegir. Por ejemplo, un clasificador del alfabeto solo producirá 26 salidas, una por cada letra.

Estas capas toman la salida producida por las capas anteriores (convoluciones, *pooling* o funciones de activación), y utilizan todos estos mapas de activación como su entrada. Por lo cual, cada mapa de activación está conectado a cada neurona de la capa totalmente conectada y multiplicada por un peso (ponderación) para producir la salida, determinando qué

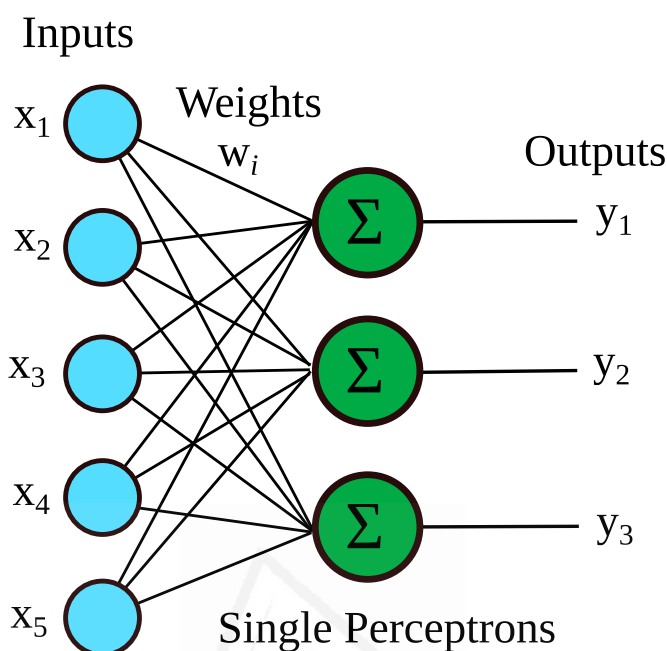


Figura A.12: Perceptrón Multicapa.

características de alto nivel están más correlacionadas con una categoría particular.

Básicamente las capas totalmente conectadas buscan correlaciones fuertes entre las características de alto nivel y una categoría particular con sus pesos particulares. Por lo tanto, una vez los productos entre las capas previas y los pesos se computen, las categorías obtendrán la probabilidad correcta.

Una capa totalmente conectada consiste de varios perceptrones multicapa. Estos se pueden describir como la función matemática que realiza un mapeado de un conjunto de valores de entrada (mapas de activación en CNNs) hacia un conjunto de valores de salida (categorías en las CNNs). Esta función está formada por la unión de muchas funciones simples. Podemos pensar que cada aplicación de una función matemática diferente como proveedora de una nueva representación de la entrada [Goodfellow et al., 2016]. En la Figura A.13 se muestra una estructura usual de capas totalmente conectadas, donde tres de estas capas se concatenan para producir la salida de la red.



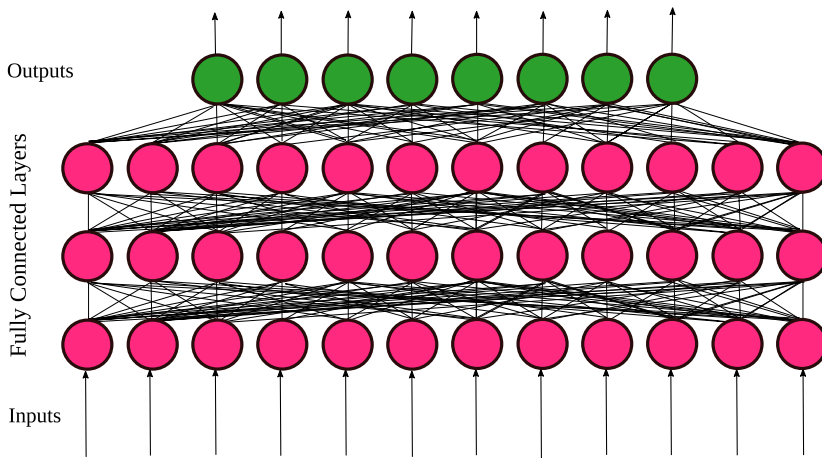


Figura A.13: Capas totalmente conectadas.

#### A.5.1.5 Capa *Softmax*

Las capas *softmax* toma la entrada generada por las capas totalmente conectadas y expresa estos valores como una distribución de probabilidad. Por lo cual, el sumatorio de los valores de salida de la red neuronal siempre será igual a 1.0.

#### A.5.1.6 Retro-propagación (*Backpropagation*)

En las CNNs las diferentes capas ocultas producen una salida que será utilizada en la siguiente capa de la arquitectura. Estas salidas son utilizadas por las neuronas de las capas para producir su propia salida. La propagación de estos resultados entre las diferentes capas genera la respuesta de la CNN. Esta respuesta es producida por multiplicaciones sucesivas de los valores de las salidas y los pesos que representan la aportación de cada neurona a la respuesta final. Sin embargo, durante la fase de entrenamiento, el algoritmo de retro-propagación es el encargado de calcular cómo de bien se ajusta la salida de la CNN a la respuesta deseada, para cada instancia. Este algoritmo calcula un valor de error para cada salida. Luego, este valor es retro-propagado a las capas ocultas de la CNN, cada capa oculta se verá afectada por el valor de error dependiendo de cómo las neuronas de la capa aportaron a la respuesta calculada. Siguiendo la misma estrate-

gia los pesos de los *kernels* en las capas convolucionales son actualizados. Este proceso asegura que las diferentes neuronas se puedan especializar en detectar diferentes características y así durante la etapa de prueba de la red, estas neuronas se activarán para una imagen nueva/desconocida [Goodfellow et al., 2016].

### A.5.1.7 *Dropout*

*Dropout* [Srivastava et al., 2014] es un método de regularización para mejorar la ejecución de las redes neuronales mediante la reducción del sobre ajuste (*overfitting*). El algoritmo de retro-propagación puede llevar a desarrollar adaptaciones que funcionen para la secuencia de entrenamiento, pero sin la capacidad de generalizar para datos nunca vistos por el modelo. El método *Dropout* propone la desactivación aleatoria de las salidas de algunas neuronas, con el objetivo de romper estas adaptaciones haciendo independiente la presencia de cualquier unidad oculta particular. *Dropout* trabaja desactivando aleatoriamente unidades y sus conexiones en la red neuronal durante el proceso de entrenamiento de la red. La idea detrás de este algoritmo es evitar que las unidades se adapten demasiado a los datos de entrenamiento. [Goodfellow et al., 2016] define este algoritmo como la función que entrena el conjunto de todas las sub-redes que se pueden formar mediante la remoción de unidades (en las capas ocultas) de una red neuronal subyacente. Este proceso se puede realizar mediante la multiplicación de una salida de una neurona por un valor de cero.

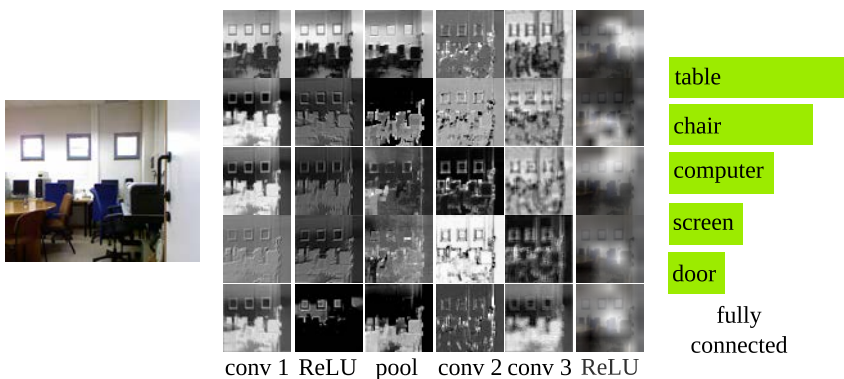
### A.5.2 Arquitecturas de CNN

Cada CNN posee su propia arquitectura y esta se puede describir como la aplicación sucesiva de diferentes filtros y capas. Cada capa tiene la capacidad de reconocer características específicas de una imagen, desde un nivel bajo (píxeles) hasta patrones más detallados (formas). Cada CNN involucra la interacción de varias de estas capas. Según su función las capas pueden ser convolucionales, de *pooling* o totalmente conectadas, entre otras. Actualmente existen un amplio número de arquitecturas diferentes. Cada una tiene ciertas propiedades distintivas, como por ejemplo, la canti-

dad de capas convolucionales. GoogLeNet [Szegedy et al., 2014] y AlexNet [Krizhevsky et al., 2012] son dos arquitecturas ampliamente conocidas para su utilización en problemas de descripción/clasificación de imágenes en el campo de las CNNs.

Un modelo CNN puede ser definido como la combinación de un arquitectura y un conjunto de datos el cual fue utilizado como conjunto de entrenamiento para dicha arquitectura. La última capa de modelo de CNN es la responsable del proceso de clasificación y esta se encarga de realizar el mapeado de los valores calculados por las capas iniciales hacia la salida de la red la cual posee un tamaño definido. Cada arquitectura debe especificar el número de salidas que generará. El tamaño de esta salida corresponde a la dimensión del problema de clasificación. Cada modelo codifica la información recolectada del conjunto de datos seleccionado como conjunto de entrenamiento, por lo que, la salida de la red estará determinada por este conjunto.

Cada capa de una CNN produce modificaciones sobre la salida de la capa previa. En la Figura A.14 se muestran las salidas producidas por algunas de las capas de una CNN, así como también, se visualizan las modificaciones producidas por cada capa. Al analizar la última capa de ReLU, es distinguible como las CNNs son capaces de resaltar las regiones de la imagen donde están presentes los objetos.



**Figura A.14:** Resultados producidos por algunas de las capas de una CNN.

## A.6 Sistemas para *Deep Learning*

En esta sección se introducen los sistema para *Deep Learning* utilizados en los experimentos durante el desarrollo de esta tesis.

### A.6.1 Clarifai

Clarifai<sup>4</sup> es uno de los sistemas conocidos de etiquetado remoto de imágenes. Específicamente cualquier imagen de entrada es etiquetada con las categorías semánticas que mejor describan el contenido de la misma. Clarifai confía en el empleo de CNNs [Krizhevsky et al., 2012] para realizar el procesamiento de una imagen y por lo tanto, generar una lista de etiquetas que describen la imagen. Este enfoque fue propuesto en primer lugar como una solución para el reto de clasificación de objetos ImageNet [Russakovsky et al., 2015] en su edición 2013 [Krizhevsky et al., 2012], donde este enfoque produjo uno de los resultados en el TOP-5 del reto. Sin embargo, el servicio de Clarifai es ahora un sistema cerrado cuyos detalles internos sobre los conjuntos de datos de entrenamiento (los cuales determinan la dimensión de la capa de clasificación) y sus arquitectura interna no son accesibles a los usuarios del sistema. Por lo tanto, estimamos el número máximo de la dimensión para los descriptores extraídos en un paso preliminar en el cual se descubren todas las anotaciones producidas por Clarifai para el conjunto de datos. Similar a la identificación del *codebook* o diccionario cuando se aplica un enfoque *Bag-of-Words* [Filliat, 2007].

Clarifai trabaja mediante el análisis de imágenes para producir una lista descriptiva de etiquetas que representan la imagen de entrada. Para cada etiqueta de la lista, el sistema incluye también un valor de probabilidad. Este valor representa cómo de probable es que una etiqueta pueda describir la imagen de entrada. El API de Clarifai puede ser accedido como un servicio web.

---

<sup>4</sup><http://www.clarifai.com>

## A.6.2 Caffe

En varias de las propuestas de esta tesis, tomamos ventaja del sistema de desarrollo Caffe, siglas de *Convolutional Architecture for Fast Feature Embedding* [Jia et al., 2014]. Este es un sistema rápido, modular y bien documentado, el cual es ampliamente utilizado por investigadores. Optamos por este entorno debido a la gran comunidad de usuarios que proveen modelos pre-entrenados que están listos para utilizar en cualquier versión de Caffe. La utilización de Caffe ha resultado en soluciones a diferentes tareas como son reconocimiento de objetos [Chatfield et al., 2014] o clasificación de escenas [Zhou et al., 2014]. Este sistema es desarrollado y mantenido por el centro de Visión y Aprendizaje de la Universidad de Berkeley (BVLC por sus siglas en inglés).

### A.6.2.1 Modelos Pre-Entrenados con Caffe

Con el objetivo de entrenar un modelo CNN, debemos seleccionar una arquitectura y un conu para utilizar como conjunto de entrenamiento. La arquitectura específica los detalles internos como el número de capas de convolución o totalmente conectadas, así como también las operaciones espaciales utilizadas en las capas de *pooling*. Por otra parte, el conjunto de entrenamiento determina el número de etiquetas léxicas empleadas para definir una imagen.

Del conjunto completo de modelos pre-entrenados disponibles en el Caffe Model Zoo<sup>5</sup>, hemos seleccionado siete diferentes candidatos los cuales se muestran en la Tabla A.2. Estos modelos difieren en sus arquitecturas, el conjunto de empleado para entrenamiento y el conjunto de etiquetas léxicas predefinidas para el modelo. Optamos por estos modelos debido a que todos están entrenados usando conjuntos de datos que están anotados con un gran número de etiquetas léxicas generalistas.

Caffe provee una forma sencilla de utilizar los modelos pre-entrenados liberados por su gran comunidad de usuarios. Tales modelos han sido entrenados con diferentes objetivos y son definidos por la combinación del conjunto de datos y la arquitectura empleada para su generación.

---

<sup>5</sup><https://github.com/BVLC/caffe/wiki/Model-Zoo>

**Cuadro A.2:** Detalles de los siete modelos CNN evaluados en la propuesta.

Nombre Modelo	ARQ CNN	CC <sup>a</sup>	CTC <sup>b</sup>	Conj Datos	#Etiq
ImageNet-AlexNet	AlexNet <sup>c</sup>	5	3	ImageNet2012 <sup>d</sup>	1,000
ImageNet-CaffeNet	AlexNet	5	3	ImageNet2012	1,000
ImageNet-GoogLeNet	GoogLeNet <sup>e</sup>	11	3	ImageNet2012	1,000
ImageNet-VGG	VGG CNN-s <sup>f</sup>	5	3	ImageNet2012	1,000
Hybrid-AlexNet	AlexNet	5	3	Hybrid MIT <sup>g</sup>	1,183
Places-AlexNet	AlexNet	5	3	Places205 MIT <sup>g</sup>	205
Places-GoogLeNet	GoogLeNet	11	3	Places205 MIT	205

<sup>a</sup> Capas Convolucionales

<sup>b</sup> Capas Totalmente Conectadas

<sup>c</sup> [Krizhevsky et al., 2012]

<sup>d</sup> [Russakovsky et al., 2015, Deng et al., 2009]

<sup>e</sup> [Szegedy et al., 2014]

<sup>f</sup> [Chatfield et al., 2014]

<sup>g</sup> [Zhou et al., 2014]

### A.6.3 Arquitecturas CNN para los modelos pre-entrenados de Caffe

#### A.6.3.1 AlexNet

La arquitectura AlexNet [Krizhevsky et al., 2012] fue desarrollada para el reto ImageNet en su edición 2012, obteniendo el primer lugar en los resultados de la competición. Esta arquitectura sigue la definición básica de una CNN, utilizando una sucesiva combinación de capas de convolución y *pooling* y un conjunto de capas totalmente conectadas al final de diseño. La arquitectura de la red se compone de cinco capas convolucionales y tres capas totalmente conectadas. AlexNet modela la salida de la red neuronal utilizando Unidades Rectificadoras Lineales (ReLU) A.5.1.2, reemplazando las funciones estándar *tanh()* o *sigmoid*. Las ReLUs mejoran la velocidad de entrenamiento de los modelos.

#### A.6.3.2 GoogLeNet

La arquitectura para CNN GoogLeNet [Szegedy et al., 2014] fue presentada para la competición ImageNet en su edición 2014, obteniendo el primer lugar en la competición y superando los resultados producidos por Clarifai en el 2013 y AlexNet en el 2012. La arquitectura incluye un nuevo

concepto en el diseño de CNNs llamado módulo *inception* (Figura A.15). Este módulo trabaja aplicando varios filtros de convolución de diferentes tamaños ( $1 \times 1$ ,  $3 \times 3$ ,  $5 \times 5$ ) a la misma entrada y en el mismo momento. Luego el resultado de cada filtro se concatena como la salida de la capa. Una capa incluye el procedimiento de *pooling* para la entrada. Este arreglo de pequeños filtros permite la utilización de menor cantidad de parámetros, lo cual mejora la ejecución del modelo. GoogLeNet está compuesto de 27 capas, de las cuales 22 tienen parámetros y las cinco restantes son capas de *pooling*. Sin embargo, el número de bloques individuales en el diseño es alrededor de 100.

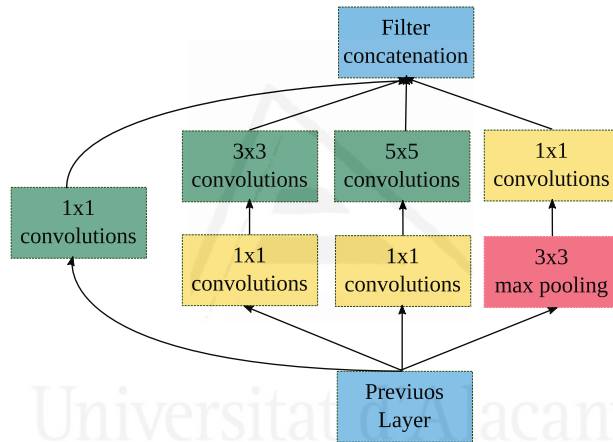


Figura A.15: Módulo *Inception*.

## A.6.4 Conjuntos de datos para los modelos Pre-entrenados de Caffe

### A.6.4.1 ImageNet 2012

El conjunto de datos ImageNet 2012 [Russakovsky et al., 2015, Deng et al., 2009] es un subconjunto de ImageNet el cual está compuesto de más de 10 millones de imágenes pertenecientes a 21,000 categorías. Este subconjunto contiene más de un millón de imágenes de 1,000 categorías diferentes, por lo cual, para cada categoría este incluye aproximadamente 1,000 imágenes. ImageNet se creó para el reto de clasificación en el ImageNet edición 2012 y ha sido utilizado en la competencia durante los

últimos años. Las categorías en el conjunto de datos incluyen objetos de diferentes tipos como: animales, instrumentos musicales y productos de aseo, entre otros. Este conjunto ha sido creado como una colección ordenada de imágenes representando objetos y escenas, jerárquicamente agrupados y ordenados por categorías. En la Tabla A.3 se muestran imágenes de cuatro categorías en el conjunto de datos ImageNet.

Cuadro A.3: Imágenes de ImageNet



#### A.6.4.2 Places 205

El conjunto de datos *Places 205* [Zhou et al., 2014] consiste en una colección de 2,488,873 imágenes agrupados en 205 categorías de escenas. Este conjunto es un subconjunto del *Places* original el cual contiene más



de siete millones de imágenes agrupadas en 476 categorías. *Places 205* es un conjunto de datos centrado en escenas que se enfoca principalmente en categorías de interiores y exteriores. *Places 205* fue creado por el MIT como una consecuencia de la falta de conjunto de datos centrados en las escenas para tareas de clasificación. Además, este conjunto fue empleado para aprender características complejas de las imágenes utilizando varias arquitecturas de CNNs para clasificar imágenes. En la Tabla A.4 se muestran imágenes de cuatro categorías en el conjunto de datos *Places 205*.

**Cuadro A.4:** Imágenes de *Places 205*



### A.6.4.3 *Hybrid* MIT

El conjunto de datos *Hybrid* o Híbrido [Zhou et al., 2014] fue creado y presentado al mismo tiempo que *Places* y por los mismos autores. Este conjunto se construyó mezclando las imágenes pertenecientes a los ImageNet 2012 y *Places* 205, obteniendo 3.5 millones de imágenes. Por lo cual, se obtienen 1,183 categorías, donde las clases repetidas en los conjuntos originales fueron fusionadas. Este conjunto de datos fue originalmente construido para aprender características complejas empleando CNNs obteniendo buenos resultados en tareas de clasificación.

## A.7 Propuesta

Después de describir la motivación de este trabajo y analizando el estado del arte en la Comprensión de Imágenes, hemos identificado un conjunto de enfoques con el objetivo de desarrollar un método robusto de comprensión de imágenes. Entre estos enfoques hemos notado un enfoque casi inexplorado en el tema de la comprensión de escenas basados en la presencia de objetos en el entorno. Consecuentemente, proponemos llevar a cabo un estudio experimental con el objetivo de encontrar una manera de describir completamente una escena considerando los objetos que se encuentran en esta. Debido a que la comprensión de escenas involucra tareas de detección y anotación de imágenes, uno de los primeros pasos será determinar el tipo de dato de entrada que se utilizará para la solución propuesta. La evaluación de datos 3D es el primer paso, para ello proponemos la utilización del algoritmo *Growing Neural Gas* (GNG) para representar el espacio de entrada de nubes de puntos con ruido y ejecutar un procedimiento de reconocimiento de objetos en nubes 3D. Basados en investigaciones previas enfocadas en *neural gases* [Garcia-Rodriguez, 2009, Angelopoulou et al., 2005], las GNGs tienen la capacidad de crecer adaptando su topología para representar información 2D, produciendo una representación más pequeña con una ligera influencia del ruido de los datos de entrada [Stergiopoulou and Papamarkos, 2006, Doucette et al., 2001, Garcia-Rodriguez et al., 2010, Baena et al., 2013]. Aplicada a datos 3D, la GNG representa un enfoque acertado para lidiar con el ruido en las nubes 3D.

Sin embargo, trabajar con datos 3D supone un conjunto de problemas como la falta de conjuntos de datos de objetos 3D con suficientes modelos para lograr generalizar los métodos a situaciones reales, así como también, el costo computacional de procesar datos tri-dimensionales es elevado y demanda grandes cantidades de espacio de almacenamiento. Los problemas mencionados nos motivaron a explorar nuevos enfoques para el desarrollo de la tarea de reconocimiento de objetos. Por lo tanto, basados en los resultados positivos que han obtenido las Redes Neuronales Convolucionales (CNNs) en las últimas ediciones del reto de reconocimiento ImageNet [Russakovsky et al., 2015], proponemos llevar a cabo una evaluación de las CNNs como un sistema de detección de objetos. Inicialmente las CNNs fueron propuestas por [LeCun et al., 1989, LeCun et al., 1990] desde la década de los 90s, estas actualmente son de fácil implementación gracias a las mejoras de *hardware*. Estas redes han sido ampliamente evaluadas para problemas de clasificación que involucran objetos, peatones, ondas de sonido, señales de tráfico e imágenes médicas entre otras.

Por otra parte, un valor agregado de las CNNs es la capacidad de descripción semántica producida por las categorías/etiquetas que la red es capaz de identificar y que se puede traducir como una explicación semántica de la imagen de entrada. Consecuentemente, proponemos la utilización de estas etiquetas semánticas como un descriptor de escenas con el fin de construir un modelo de clasificación supervisado. Además, también proponemos la utilización de este descriptor semántico para la solución de retos de mapeado y la evaluación de las capacidades descriptivas de las etiquetas léxicas.

Además, el descriptor semántico puede ser adecuado para la clasificación no supervisada de los lugares o entornos, por lo cual, proponemos su utilización en este tipo de problemas con el objetivo de lograr un método de clasificación robusto de escenas. Finalmente, para tratar con el problema de reconocimiento de objetos, proponemos desarrollar un estudio experimental para la clasificación no supervisada de los objetos presentes en una nube de puntos y usar estos como las instancias de entrenamiento de un clasificador.

## A.8 Objetivos

El objetivo principal de esta investigación es el desarrollo y validación de un método eficiente de comprensión de escenas basado en objetos, que será capaz de ayudar a resolver los problemas relacionados con la identificación de lugares para robots móviles. Buscamos analizar las propuestas en el estado del arte actual con el fin de encontrar el método que mejor se ajuste a nuestros objetivos, así como también seleccionar el tipo de datos más conveniente para tratar con el problema.

Con el fin de identificar objetos con datos 3D, planeamos evaluar la representación de objetos con la GNG para afrontar los problemas derivados del ruido en las nubes 3D. Con respecto a los datos 2D, experimentaremos empleando técnicas de *deep learning* tomando ventaja de sus capacidades de generalización. En vista de los puntos arriba mencionados definimos, como un objetivo primario, encontrar una representación precisa para las escenas mediante el uso de etiquetas semánticas o con descriptores de nubes de puntos 3D.

Como objetivo secundario mostraremos la bondad de la utilización de descriptores semánticos generados con modelos pre-entrenados para problemas de mapeado y clasificación de escenas, así como también el uso de modelos de *deep learning* en conjunto con procedimientos de descripción 3D para construir un modelo de clasificación de objetos 3D lo cual está directamente relacionado con el objetivo de representación de datos de este trabajo.

## A.9 Conclusiones

En esta tesis nos enfocamos en resolver problemas de comprensión de escenas para robots móviles. Este tipo de problema está usualmente dirigido mediante procedimientos de reconocimiento de objetos. Por lo cual, como primer paso hemos presentado un estudio para evaluar cómo manejar el ruido en nubes de puntos 3D con el objetivo de identificar objetos. En este estudio, se utilizaron las *Growing Neural Gas* (GNG) para representar nubes con ruido y después realizar una comparación con otros

métodos de representación 3D. Sin embargo, a pesar de que la GNG produjo resultados positivos, el reconocimiento de objetos 3D todavía enfrenta varios problemas así como también elevados requerimientos computacionales. Consecuentemente, decidimos explorar el empleo de información 2D en el procedimiento.

Además, después de estudiar los enfoques de reconocimiento 2D, definimos un nuevo método para representar imágenes, basado en etiquetas semánticas, generadas con un modelo de Red Neuronal Convolutiva (CNN). Estas etiquetas semánticas incluyen también un valor de confianza representando cómo de probable es para una etiqueta estar presente en la imagen. Este par de valores (etiqueta-confianza) serán utilizados como un descriptor para las imágenes. El método propuesto emplea técnicas de *deep learning*, pero no requiere de largos periodos de entrenamiento para construir un modelo de clasificación/etiquetado. Por lo tanto, el método propuesto se puede adaptar a cualquier sistema de etiquetado que produzca una etiqueta con su correspondiente valor de confianza.

El descriptor semántico nos permite trabajar con imágenes obteniendo medidas para comparación entre imágenes con el fin de agruparlas basado en su similitud. Empleando este descriptor, podemos inferir información semántica de una imagen o de un grupo de ellas. La información semántica puede describir la composición de las imágenes (presencia de objetos) así como también su probable categoría semántica (habitación, cocina). Por consiguiente, los descriptores semánticos contribuyen no solo como descriptores regulares de imágenes sino también con sus capacidades de descripción semántica.

En adición, hemos evaluado el descriptor semántico en problemas de clasificación supervisada para escenas de interior. Los resultados demuestran la bondad del descriptor semántico cuando es comparado con descriptores numéricos tradicionales.

Además, el descriptor semántico fue evaluado en problemas de mapeado Semántico y Topológico. En estos problemas el uso de una representación precisa de la imagen robustece los mapas construidos. En el mapeado el descriptor semántico suple el procedimiento con una amplia descripción de los objetos que posiblemente se encuentre en el entorno. Además, mediante

el mapeado semántico la categoría de las ubicaciones puede ser derivada solo tomando en cuenta las etiquetas con los mayores valores medios de probabilidad, los cuales se obtuvieron mediante el agrupamiento de imágenes similares en una secuencia. De igual manera, este procedimiento de inferencia puede ser aplicado a los mapas topológicos, en cuyo caso los grupos corresponden a nodos de imágenes similares en una secuencia.

Por último, pero no menos importante, los modelos CNN han sido evaluados en un sistema de etiquetado de grupos de puntos segmentados de nubes de puntos 3D, con el objetivo de entrenar un clasificador que pueda reconocer objetos 3D que generalmente son difíciles de detectar cuando se emplean clasificadores 2D de imágenes.

Finalmente, los estudios llevados a cabo en esta tesis nos permiten confirmar que utilizando los descriptores semánticos producidos por una herramienta externa de etiquetado, en nuestro caso modelos CNN, permiten obtener un alto nivel de comprensión de escenas, así como también su significado semántico.

## A.10 Contribuciones de la Tesis

Las contribuciones realizadas durante el desarrollo de esta investigación son las siguientes:

- La aplicación de un algoritmo basado en GNG para representar escenas con ruido y evaluar el reconocimiento de objetos en estas escenas.
- El uso de modelos CNN pre-entrenados como sistema de etiquetado para construir un descriptor semántico de imágenes.
- La aplicación del descriptor semántico en procedimientos de mapeado con el objetivo de inferir la información de los lugares.
- Lo construcción de clasificadores de objetos 3D usando objetos etiquetados con un modelo CNN pre-entrenado para reconocimiento de imágenes 2D.

## A.11 Publicaciones

Los siguientes artículos fueron publicados como resultados de las investigaciones llevadas a cabo por el candidato doctoral:

- Artículos Publicados en Revistas Científicas:
  - José Carlos Rangel, Jesús Martínez-Gómez, Cristina Romero-González, Ismael García-Varea and Miguel Cazorla: OReSLab: 3D Object Recognition through CNN Semi-Supervised Labeling. Under Review. **Applied Soft Computing** Factor de Impacto (JCR 2015): 2.857.
  - José Carlos Rangel, Miguel Cazorla, Ismael García-Varea, Cristina Romero-González and Jesús Martínez-Gómez: AuSeMap: Automatic Semantic Map Generation. Under Review. **Autonomous Robots** Factor de Impacto (JCR 2015): 1.547.
  - José Carlos Rangel, Jesús Martínez-Gómez, Ismael García-Varea y Miguel Cazorla: LexToMap: Lexical-based Topological Mapping. **Advanced Robotics** 31(5):268-281 (2017) Factor de Impacto (JCR 2015): 0.516.
  - José Carlos Rangel, Miguel Cazorla, Ismael García-Varea, Jesús Martínez-Gómez, Elisa Fromont y Marc Sebban: Scene Classification based on Semantic Labeling. **Advanced Robotics** 30(11-12):758-769 (2016) Factor de Impacto (JCR 2015): 0.516.
  - José Carlos Rangel, Vicente Morell, Miguel Cazorla, Sergio Orts-Escolano y José García-Rodríguez: Object recognition in noisy RGB-D data using GNG. **Pattern Analysis and Applications** Aceptado (2016) Factor de Impacto (JCR 2015): 1.104.

- Conferencias Internacionales:
  - José Carlos Rangel , Miguel Cazorla, Ismael García-Varea, Jesús Martínez-Gómez, Éliisa Fromont and Marc Sebban: **Computing Image Descriptors from Annotations Acquired from External Tools**. Robot 2015: Second Iberian Robotics Conference, Lisbon, Portugal: 673-683, November 19-21.
  - José Carlos Rangel, Vicente Morell, Miguel Cazorla, Sergio Orts-Escolano and José García-Rodríguez: **Using gng on 3d object recognition in noisy rgb-d data**. International Joint Conference on Neural Networks, IJCNN 2015, Killarney, Ireland, July 12-17.
  - José Carlos Rangel, Vicente Morell, Miguel Cazorla, Sergio Orts-Escolano and José García-Rodríguez: **Object Recognition in Noisy RGB-D Data**. International work-conference on the interplay between natural and artificial computation, IWINAC 2015, Elche, Spain: 261-270, June 1-5.

## A.12 Trabajo Futuro

Como trabajo futuro proponemos la integración de los métodos de clasificación semántica desarrollados en una plataforma robótica móvil, con el objetivo de lograr una plataforma móvil que se ejecute en tiempo real destinada al análisis de estancias, la cual pueda ser utilizada en proyectos de asistencia al hogar.

En lo relativo al descriptor semántico, se plantea seguir estudiando sus capacidades descriptivas mediante la selección de etiquetas léxicas sobresalientes, tomando en cuenta la categoría real a la que pertenece una imagen.

Se propone también el empleo de modelos en conjunto, es decir la aplicación de varios modelos a un mismo conjunto de imágenes. Dichos mo-



delos se centrarán cada uno en diferentes aspectos, como escenas, objetos, instrumentos, mobiliario, etc., con el objetivo de obtener la clasificación semántica de un lugar así como también su posible contenido y de esta manera fortalecer la comprensión de la escena.

Proponemos también el estudio de las facultades que tiene el descriptor semántico para ser empleados en escenas de exterior, donde se enfrentan desafíos como una perspectiva mayor del entorno, mayor número de elementos en las escenas, diversos tipos y fuentes de iluminación, así como también elementos que pueden aparecer en tamaños imperceptibles para la red neuronal.

Para la generación de descriptores semánticos planteamos la utilización de nuevos diseños de Arquitecturas, como ResNet, modelos pre-entrenados y sistemas de desarrollo como Keras con Tensor Flow, para completar un estudio comparativo de estos con los modelos empleados en esta tesis con el objetivo de lograr métricas de evaluación que permitan seleccionar el modelo que más se ajuste a un tipo de problema de estudio.

# Bibliography

---

- [Abadi et al., 2015] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org. 130
- [Aldoma et al., 2012] Aldoma, A., Tombari, F., Di Stefano, L., and Vincze, M. (2012). A global hypotheses verification method for 3d object recognition. In Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., and Schmid, C., editors, *Computer Vision ECCV 2012*, volume 7574 of *Lecture Notes in Computer Science*, pages 511–524. Springer Berlin Heidelberg. 33, 38, 43, 126, 128
- [Angeli et al., 2009] Angeli, A., Doncieux, S., Meyer, J.-A., and Filliat, D. (2009). Visual topological SLAM and global localization. In *International Conference on Robotics and Automation*, pages 4300–4305, Kobe, Japan. IEEE. 83
- [Angelopoulou et al., 2005] Angelopoulou, A., Psarrou, A., Rodríguez, J., and Revett, K. (2005). Automatic Landmarking of 2D Medical Shapes Using the Growing Neural Gas Network. In Liu, Y., Jiang, T., and Zhang, C., editors, *Computer Vision for Biomedical Image Applica-*

- tions, volume 3765 of *Lecture Notes in Computer Science*, pages 210–219. Springer Berlin / Heidelberg. 27, 183
- [Asari et al., 2014] Asari, M., Sheikh, U., and Supriyanto, E. (2014). 3d shape descriptor for object recognition based on kinect-like depth image. *Image and Vision Computing*, 32(4):260 – 269. 33, 126, 128
- [Baena et al., 2013] Baena, R. M. L., López-Rubio, E., Domínguez, E., Palomo, E. J., and Jerez, J. M. (2013). A self-organizing map for traffic flow monitoring. In *IWANN (2)*, pages 458–466. 27, 183
- [Bai et al., 2016] Bai, S., Bai, X., Zhou, Z., Zhang, Z., and Latecki, L. J. (2016). Gift: A real-time and scalable 3d shape search engine. *arXiv preprint arXiv:1604.01879*. 129
- [Ball et al., 2013] Ball, D., Heath, S., Wiles, J., Wyeth, G., Corke, P., and Milford, M. (2013). Openratslam: an open source brain-based slam system. *Autonomous Robots*, 34(3):149–176. 85
- [Bay et al., 2006] Bay, H., Tuytelaars, T., and Van Gool, L. (2006). SURF: Speeded up robust features. In *Computer vision–ECCV 2006*, pages 404–417. Springer. 83
- [Bengio, 2009] Bengio, Y. (2009). Learning deep architectures for AI. *Foundations and trends® in Machine Learning*, 2(1):1–127. 86
- [Besl and McKay, 1992] Besl, P. and McKay, N. (1992). A method for registration of 3-d shapes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14(2):239–256. 43
- [Bhattacharya and Gavrilova, 2008] Bhattacharya, P. and Gavrilova, M. L. (2008). Roadmap-based path planning - using the voronoi diagram for a clearance-based shortest path. *IEEE Robotics Automation Magazine*, 15(2):58–66. 82
- [Bilen et al., 2014] Bilen, H., Pedersoli, M., and Tuytelaars, T. (2014). Weakly supervised object detection with posterior regularization. In *BMVC*. 132

- [Bilen et al., 2015] Bilen, H., Pedersoli, M., and Tuytelaars, T. (2015). Weakly supervised object detection with convex clustering. In *CVPR*. 132
- [Blanco et al., 2007] Blanco, J., Fernández-Madrigal, J., and Gonzalez, J. (2007). A new approach for large-scale localization and mapping: Hybrid metric-topological slam. In *International Conference on Robotics and Automation*, pages 2061–2067. IEEE. 105
- [Bo et al., 2013] Bo, L., Ren, X., and Fox, D. (2013). Unsupervised feature learning for rgb-d based object recognition. In *Experimental Robotics*, pages 387–402. Springer. 11, 85, 166
- [Booiij et al., 2007] Booiij, O., Terwijn, B., Zivkovic, Z., and Kröse, B. (2007). Navigation using an appearance based topological map. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 3927–3932. IEEE. 82
- [Bosch et al., 2007] Bosch, A., Zisserman, A., and Munoz, X. (2007). Representing shape with a spatial pyramid kernel. In *Proceedings of the 6th ACM International Conference on Image and Video Retrieval, CIVR '07*, pages 401–408, New York, NY, USA. ACM. 64
- [Breiman, 2001] Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32. 69
- [Brock et al., 2016] Brock, A., Lim, T., Ritchie, J., and Weston, N. (2016). Generative and discriminative voxel modeling with convolutional neural networks. *arXiv preprint arXiv:1608.04236*. 130
- [Bui et al., 2017] Bui, H. M., Lech, M., Cheng, E., Neville, K., and Burnett, I. S. (2017). Object recognition using deep convolutional features transformed by a recursive network structure. *IEEE Access*, PP(99):1–1. 126, 131
- [Burgard et al., 2009] Burgard, W., Stachniss, C., Grisetti, G., Steder, B., Kümmerle, R., Dornhege, C., Ruhnke, M., Kleiner, A., and Tardós, J. (2009). A comparison of slam algorithms based on a graph of relations.

- In *International Conference on intelligent Robots and Systems*, pages 2089–2095. IEEE. 105
- [Bylow et al., 2013] Bylow, E., Sturm, J., Kerl, C., Kahl, F., and Cremers, D. (2013). Real-time camera tracking and 3d reconstruction using signed distance functions. 109
- [Carneiro et al., 2015] Carneiro, G., Nascimento, J., and Bradley, A. P. (2015). Unregistered multiview mammogram analysis with pre-trained deep learning models. In Navab, N., Hornegger, J., Wells, W. M., and Frangi, A. F., editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III*, pages 652–660. Springer International Publishing, Cham. 12, 85, 167
- [Chang and Lin, 2011] Chang, C.-C. and Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27:1–27:27. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>. 69, 138
- [Chatfield et al., 2014] Chatfield, K., Simonyan, K., Vedaldi, A., and Zisserman, A. (2014). Return of the devil in the details: Delving deep into convolutional nets. In *Proceedings of the British Machine Vision Conference*, Nottingham, UK. BMVA Press. 22, 23, 178, 179
- [Chen and Bhanu, 2007] Chen, H. and Bhanu, B. (2007). 3d free-form object recognition in range images using local surface patches. *Pattern Recognition Letters*, 28(10):1252 – 1262. 43
- [Chen and Medioni, 1991] Chen, Y. and Medioni, G. (1991). Object modeling by registration of multiple range images. In Medioni, G., editor, *1991 Proceedings., IEEE International Conference on Robotics and Automation, 1991.*, pages 2724–2729 vol.3. 43
- [Chen et al., 2014] Chen, Z., Lam, O., Jacobson, A., and Milford, M. (2014). Convolutional neural network-based place recognition. *CoRR*, abs/1411.1509. 107

- [Choset and Nagatani, 2001] Choset, H. and Nagatani, K. (2001). Topological simultaneous localization and mapping (slam): toward exact localization without explicit localization. *IEEE Transactions on Robotics and Automation*, 17(2):125–137. 106
- [Chrisman, 1992] Chrisman, L. (1992). Reinforcement learning with perceptual aliasing: the perceptual distinctions approach. In *Proceedings of the tenth national conference on Artificial intelligence*, pages 183–188, San Jose, California, USA. AAAI Press. 86
- [Clarifai, 2015] Clarifai (2015). Clarifai: Amplifying intelligence. 13, 67, 168
- [Computer Vision LAB, 2013] Computer Vision LAB (2013). SHOT: Unique signatures of histograms for local surface description - computer vision LAB. 6, 41, 161
- [Cover and Hart, 1967] Cover, T. and Hart, P. (1967). Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on*, 13(1):21–27. 69
- [Csurka et al., 2004] Csurka, G., Dance, C., Fan, L., Willamowski, J., and Bray, C. (2004). Visual categorization with bags of keypoints. In *Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22. 61, 63
- [Cummins and Newman, 2008] Cummins, M. and Newman, P. (2008). FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance. *The International Journal of Robotics Research*, 27(6):647–665. 84
- [Cyr and Kimia, 2001] Cyr, C. M. and Kimia, B. B. (2001). 3D object recognition using shape similarity-based aspect graph. In *International Conference on Computer Vision*, volume 1, pages 254–261, Vancouver, British Columbia, Canada. IEEE. 84
- [Dai et al., 2016] Dai, A., Nießner, M., Zollöfer, M., Izadi, S., and Theobalt, C. (2016). BundleFusion: Real-time Globally Consistent 3D

- Reconstruction using On-the-fly Surface Re-integration. *arXiv preprint arXiv:1604.01093*. 109
- [Dalal and Triggs, 2005] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1*, CVPR '05, pages 886–893, Washington, DC, USA. IEEE Computer Society. 65, 83
- [Demšar, 2006] Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7:1–30. 70
- [Deng et al., 2009] Deng, J., Dong, W., Socher, R., Li, L., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE. 23, 24, 83, 127, 130, 131, 179, 180
- [Doucette et al., 2001] Doucette, P., Agouris, P., Stefanidis, A., and Musavi, M. (2001). Self-organised clustering for road extraction in classified imagery. *{ISPRS} Journal of Photogrammetry and Remote Sensing*, 55(5–6):347–358. 27, 183
- [Endres et al., 2012] Endres, F., Hess, J., Engelhard, N., Sturm, J., Cremers, D., and Burgard, W. (2012). An evaluation of the rgb-d slam system. In *International Conference on Robotics and Automation*, pages 1691–1696. IEEE. 108
- [Filliat, 2007] Filliat, D. (2007). A visual bag of words method for interactive qualitative localization and mapping. In *International Conference on Robotics and Automation*, pages 3921–3926, Roma, Italy. IEEE. 21, 84, 177
- [Fraundorfer et al., 2007] Fraundorfer, F., Engels, C., and Nistér, D. (2007). Topological mapping, localization and navigation using image collections. In *International Conference on Intelligent Robots and Systems*, pages 3872–3877, San Diego, California, USA. IEEE. 82, 84

- [Friedman, 1940] Friedman, M. (1940). A comparison of alternative tests of significance for the problem of  $m$  rankings. *The Annals of Mathematical Statistics*, 11(1):86–92. 70
- [Fritzke, 1995] Fritzke, B. (1995). A growing neural gas network learns topologies. In *Advances in Neural Information Processing Systems 7*, pages 625–632. MIT Press. 32, 34
- [Fuentes-Pacheco et al., 2012] Fuentes-Pacheco, J., Ruiz-Ascencio, J., and Rendón-Mancha, J. (2012). Visual simultaneous localization and mapping: a survey. *Artificial Intelligence Review*, 43(1):55–81. 106
- [Galindo et al., 2005] Galindo, C., Saffiotti, A., Coradeschi, S., Buschka, P., Fernandez-Madrigal, J. A., and Gonzalez, J. (2005). Multi-hierarchical semantic maps for mobile robotics. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2278–2283. 106
- [Garcia-Garcia et al., 2016] Garcia-Garcia, A., Gomez-Donoso, F., Garcia-Rodriguez, J., Orts-Escolano, S., Cazorla, M., and Azorin-Lopez, J. (2016). Pointnet: A 3d convolutional neural network for real-time object class recognition. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 1578–1584. 129
- [Garcia-Rodriguez, 2009] Garcia-Rodriguez, J. (2009). *Self-organizing neural network model to represent objects and their movement in realistic scenes*. PhD thesis, University of Alicante, Alicante, Spain. 27, 183
- [Garcia-Rodriguez et al., 2010] Garcia-Rodriguez, J., Angelopoulou, A., Garcia-Chamizo, J. M., and Psarrou, A. (2010). Gng based surveillance system. In *Proc. Int Neural Networks (IJCNN) Joint Conf*, pages 1–8. 27, 183
- [Girshick, 2015] Girshick, R. (2015). Fast r-cnn. In *The IEEE International Conference on Computer Vision (ICCV)*. 126, 131



- [Girshick et al., 2014] Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR '14*, pages 580–587, Washington, DC, USA. IEEE Computer Society. 126, 131
- [Goedemé et al., 2005] Goedemé, T., Tuytelaars, T., Van Gool, L., Vanacker, G., and Nuttin, M. (2005). Feature based omnidirectional sparse visual path following. In *International Conference on Intelligent Robots and Systems*, pages 1806–1811, Edmonton, Alberta, Canada. IEEE. 84
- [Gomez-Donoso et al., 2017] Gomez-Donoso, F., Garcia-Garcia, A., Orts-Escolano, S., Garcia-Rodriguez, J., and Cazorla, M. (2017). Lonchanet: A sliced-based cnn architecture for real-time 3d object recognition. In *2017 International Joint Conference on Neural Networks (IJCNN)*. 130
- [Goodfellow et al., 2016] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. <http://www.deeplearningbook.org>. 18, 19, 173, 175
- [Guo et al., 2014] Guo, Y., Bennamoun, M., Sohel, F., Lu, M., and Wan, J. (2014). 3d object recognition in cluttered scenes with local surface features: A survey. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(11):2270–2287. 32, 40, 41, 43, 128
- [Handa et al., 2016] Handa, A., Pătrăucean, V., Stent, S., and Cipolla, R. (2016). Scenenet: An annotated model generator for indoor scene understanding. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5737–5743. 5, 159
- [Hegde and Zadeh, 2016] Hegde, V. and Zadeh, R. (2016). Fusionnet: 3d object classification using multiple data representations. *CoRR*, abs/1607.05695. 130
- [Hinterstoisser et al., 2011] Hinterstoisser, S., Holzer, S., Cagniart, C., Ilic, S., Konolige, K., Navab, N., and Lepetit, V. (2011). Multimodal

- templates for real-time detection of texture-less objects in heavily cluttered scenes. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 858–865. 33, 128
- [Jia et al., 2014] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*. 11, 22, 85, 86, 127, 130, 166, 178
- [Johns et al., 2016] Johns, E., Leutenegger, S., and Davison, A. J. (2016). Pairwise decomposition of image sequences for active multi-view recognition. *arXiv preprint arXiv:1605.08359*. 129
- [Johnson and Hebert, 1998] Johnson, A. and Hebert, M. (1998). Surface matching for object recognition in complex three-dimensional scenes. *Image and Vision Computing*, 16(9-10):635 – 651. 41
- [Johnson and Hebert, 1999] Johnson, A. and Hebert, M. (1999). Using spin images for efficient object recognition in cluttered 3d scenes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(5):433–449. 41
- [Kohonen, 1995] Kohonen, T. (1995). *Self-Organising Maps*. Springer-Verlag. 34
- [Koseck and Li, 2004] Koseck, J. and Li, F. (2004). Vision based topological markov localization. In *International Conference on Robotics and Automation*, volume 2, pages 1481–1486, New Orleans, LA, USA. IEEE. 84
- [Kostavelis and Gasteratos, 2013] Kostavelis, I. and Gasteratos, A. (2013). Learning spatially semantic representations for cognitive robot navigation. *Robotics and Autonomous Systems*, 61(12):1460 – 1475. 106
- [Kostavelis and Gasteratos, 2015] Kostavelis, I. and Gasteratos, A. (2015). Semantic mapping for mobile robotics tasks: A survey. *Robotics and Autonomous Systems*, 66:86 – 103. 106

- [Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105. 12, 13, 20, 21, 23, 85, 107, 114, 167, 168, 176, 177, 179
- [Kuipers et al., 2004] Kuipers, B., Modayil, J., Beeson, P., MacMahon, M., and Savelli, F. (2004). Local metrical and global topological maps in the hybrid spatial semantic hierarchy. In *International Conference on Robotics and Automation*, volume 5, pages 4845–4851, New Orleans, LA, USA. IEEE. 84
- [Labbé and Michaud, 2011] Labbé, M. and Michaud, F. (2011). Memory management for real-time appearance-based loop closure detection. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1271–1276. 108
- [Lampert et al., 2014] Lampert, C. H., Nickisch, H., and Harmeling, S. (2014). Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(3):453–465. 62
- [Lazebnik et al., 2006] Lazebnik, S., Schmid, C., and Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2, CVPR '06*, pages 2169–2178, Washington, DC, USA. IEEE Computer Society. 61, 65, 87
- [LeCun et al., 1989] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551. 28, 184
- [LeCun et al., 1990] LeCun, Y., Boser, B. E., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. E., and Jackel, L. D. (1990). Handwritten digit recognition with a back-propagation network. In Touret-

- zky, D. S., editor, *Advances in Neural Information Processing Systems 2*, pages 396–404. Morgan-Kaufmann. 28, 184
- [LeCun et al., 2010] LeCun, Y., Kavukcuoglu, K., and Farabet, C. (2010). Convolutional networks and applications in vision. In *International Symposium on Circuits and Systems (ISCAS 2010)*, pages 253–256, Paris, France. 12, 167
- [Lee et al., 2009] Lee, H., Grosse, R., Ranganath, R., and Ng, A. Y. (2009). Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *International Conference on Machine Learning*, pages 609–616. ACM. 86
- [Lee et al., 2005] Lee, S., Xin, J., and Westland, S. (2005). Evaluation of image similarity by histogram intersection. *Color Research & Application*, 30(4):265–274. 88
- [Lemaire et al., 2007] Lemaire, T., Berger, C., Jung, I., and Lacroix, S. (2007). Vision-based slam: Stereo and monocular approaches. *International Journal of Computer Vision*, 74(3):343–364. 106
- [Leonard and Durrant-Whyte, 1991] Leonard, J. and Durrant-Whyte, H. (1991). Mobile robot localization by tracking geometric beacons. *Robotics and Automation, IEEE Transactions on*, 7(3):376–382. 3, 157
- [Li et al., 2010] Li, L., Su, H., Lim, Y., and Li, F. (2010). Objects as attributes for scene classification. In *Trends and Topics in Computer Vision - ECCV 2010 Workshops*, pages 57–69, Heraklion, Crete, Greece, September 10-11, 2010, Revised Selected Papers, Part I. 62
- [Li et al., 2014] Li, L., Su, H., Lim, Y., and Li, F. (2014). Object bank: An object-level image representation for high-level visual recognition. *International Journal of Computer Vision*, 107(1):20–39. 62
- [Li et al., 2015] Li, L., Yan, C. C., Ji, W., Chen, B.-W., Jiang, S., and Huang, Q. (2015). Lsh-based semantic dictionary learning for large scale image understanding. *Journal of Visual Communication and Image Representation*, 31:231 – 236. 5, 160

- [Li et al., 2009] Li, L. J., Socher, R., and Fei-Fei, L. (2009). Towards total scene understanding: Classification, annotation and segmentation in an automatic framework. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2036–2043. 4, 158
- [Liao et al., 2016] Liao, Y., Kodagoda, S., Wang, Y., Shi, L., and Liu, Y. (2016). Understand scene categories by objects: A semantic regularized scene classifier using convolutional neural networks. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2318–2325. 4, 158
- [Liu et al., 2009] Liu, M., Scaramuzza, D., Pradalier, C., Siegwart, R., and Chen, Q. (2009). Scene recognition with omnidirectional vision for topological map using lightweight adaptive descriptors. In *International Conference on Intelligent Robots and Systems*, pages 116–121, St. Louis, MO, USA. IEEE. 84
- [Liu and Von Wichert, 2013] Liu, Z. and Von Wichert, G. (2013). Applying rule-based context knowledge to build abstract semantic maps of indoor environments. In *Intelligent Robots and Systems (IROS), 2013*, pages 5141–5147. 106
- [Lowe, 1999] Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *The proceedings of the seventh IEEE international conference on Computer vision*, volume 2, pages 1150–1157, Kerkyra, Greece. IEEE. 83
- [Luo et al., 2007] Luo, J., Pronobis, A., Caputo, B., and Jensfelt, P. (2007). Incremental learning for place recognition in dynamic environments. In *International Conference on Intelligent Robots and Systems*, pages 721–728. IEEE. 7, 161
- [Maddern et al., 2012] Maddern, W., Milford, M., and Wyeth, G. (2012). Cat-slam: probabilistic localisation and mapping using a continuous appearance-based trajectory. *The International Journal of Robotics Research*, 31(4):429–451. 108

- [Marius Muja, 2008] Marius Muja (2008). FLANN - fast library for approximate nearest neighbors : FLANN - FLANN browse. 42
- [Maron and Ratan, 1998] Maron, O. and Ratan, A. L. (1998). Multiple-instance learning for natural scene classification. In *Proceedings of the Fifteenth International Conference on Machine Learning, ICML'98*, pages 341–349, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc. 60
- [Martinetz, 1993] Martinetz, T. (1993). Competitive Hebbian Learning Rule Forms Perfectly Topology Preserving Maps. In Gielen, S. and Kappen, B., editors, *Proc. ICANN'93, Int. Conf. on Artificial Neural Networks*, pages 427–434, London, UK. Springer. 34
- [Martinetz and Schulten, 1994] Martinetz, T. and Schulten, K. (1994). Topology representing networks. *Neural Networks*, 7(3). 34
- [Martínez-Gómez et al., 2015] Martínez-Gómez, J., Caputo, B., Cazorla, M., Christensen, H., Fornoni, M., García-Varea, I., and Pronobis, A. (2015). The robot vision challenge. where are we after 5 editions? *IEEE Robotics and Automation Magazine*. 11, 165
- [Martínez-Gomez et al., 2015] Martínez-Gomez, J., Cazorla, M., García-Varea, I., and Morell, V. (2015). ViDRILO: The Visual and Depth Robot Indoor Localization with Objects information dataset. *International Journal of Robotics Research*, 34(14):1681–1687. 9, 60, 64, 104, 112
- [Martínez-Gómez et al., 2014] Martínez-Gómez, J., Fernández-Caballero, A., García-Varea, I., Rodríguez, L., and Romero-González, C. (2014). A taxonomy of vision systems for ground mobile robots. *International Journal of Advanced Robotic Systems*, 11:1–11. 60
- [Martínez-Gómez et al., 2011] Martínez-Gómez, J., Jiménez-Picazo, A., Gamez, J. A., and García-Varea, I. (2011). Combining invariant features and localization techniques for visual place classification: successful experiences in the RobotVision@ImageCLEF competition. *Journal of Physical Agents*, 5(1):45–54. 84

- [Martínez-Gómez et al., 2016] Martínez-Gómez, J., Morell, V., Cazorla, M., and García-Varea, I. (2016). Semantic localization in the PCL library. *Robotics and Autonomous Systems*, 75, Part B:641 – 648. 87
- [Maturana and Scherer, 2015] Maturana, D. and Scherer, S. (2015). Voxnet: A 3d convolutional neural network for real-time object recognition. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 922–928. IEEE. 129
- [Meshgi and Ishii, 2015] Meshgi, K. and Ishii, S. (2015). Expanding histogram of colors with gridding to improve tracking accuracy. In *International Conference on Machine Vision Applications*, pages 475–479. IEEE. 111
- [Morell et al., 2014] Morell, V., Cazorla, M., Orts-Escolano, S., and Garcia-Rodriguez, J. (2014). 3D Maps Representation using GNG. In *Neural Networks (IJCNN), The 2014 International Joint Conference on*. 32
- [Muja and Lowe, 2014] Muja, M. and Lowe, D. G. (2014). Scalable nearest neighbor algorithms for high dimensional data. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36. 42
- [Mur-Artal et al., 2015] Mur-Artal, R., Montiel, J. M. M., and Tardós, J. D. (2015). Orb-slam: A versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163. 85
- [Murthy et al., 2015] Murthy, V. N., Maji, S., and Manmatha, R. (2015). Automatic image annotation using deep learning representations. In *International Conference on Multimedia Retrieval*, pages 603–606, Shanghai, China. ACM. 12, 85, 167
- [Neverova et al., 2014] Neverova, N., Wolf, C., Taylor, G. W., and Nebout, F. (2014). Multi-scale deep learning for gesture detection and localization. In *Computer Vision-ECCV 2014 Workshops*, pages 474–490, Zurich, Switzerland. Springer. 11, 85, 166

- [Nguyen et al., 2016] Nguyen, T. V., Liu, L., and Nguyen, K. (2016). Exploiting generic multi-level convolutional neural networks for scene understanding. In *2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pages 1–6. 4, 159
- [Nieto et al., 2006] Nieto, J., Guivant, J., and Nebot, E. (2006). Denseslam: Simultaneous localization and dense mapping. *The International Journal of Robotics Research*, 25(8):711–744. 109
- [Oliva and Torralba, 2001] Oliva, A. and Torralba, A. (2001). Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175. 64
- [Pang and Neumann, 2013] Pang, G. and Neumann, U. (2013). Training-based object recognition in cluttered 3d point clouds. In *3D Vision - 3DV 2013, 2013 International Conference on*, pages 87–94. 33, 128
- [Patterson et al., 2014] Patterson, G., Xu, C., Su, H., and Hays, J. (2014). The sun attribute database: Beyond categories for deeper scene understanding. *International Journal of Computer Vision*, 108(1-2):59–81. 5, 159
- [Paul and Newman, 2010] Paul, R. and Newman, P. (2010). Fab-map 3d: Topological mapping with spatial and visual appearance. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 2649–2656. 108, 109
- [PCL, 2011] PCL (2011). Documentation - point cloud library (PCL). 40
- [Pronobis et al., 2010] Pronobis, A., Mozos, O. M., Caputo, B., and Jensfelt, P. (2010). Multi-modal semantic place classification. *The International Journal of Robotics Research (IJRR), Special Issue on Robotic Vision*, 29(2-3):298–320. 82, 106
- [Qu et al., 2017] Qu, L., He, S., Zhang, J., Tian, J., Tang, Y., and Yang, Q. (2017). Rgb-d salient object detection via deep fusion. *IEEE Transactions on Image Processing*, PP(99):1–1. 132



- [Rangel et al., 2016a] Rangel, J., Cazorla, M., García-Varea, I., Martínez-Gómez, J., Fromont, E., and Sebban, M. (2016a). Scene classification based on semantic labeling. *Advanced Robotics*, 30(11-12):758–769. 11, 12, 85, 108, 166, 167
- [Rangel et al., 2017] Rangel, J. C., Martínez-Gómez, J., García-Varea, I., and Cazorla, M. (2017). Lextomap: lexical-based topological mapping. *Advanced Robotics*, 31(5):268–281. 108
- [Rangel et al., 2016b] Rangel, J. C., Morell, V., Cazorla, M., Orts-Escolano, S., and García-Rodríguez, J. (2016b). Object recognition in noisy rgb-d data using gng. *Pattern Analysis and Applications*, pages 1–16. 126, 128
- [Redondo-Cabrera et al., 2014] Redondo-Cabrera, C., López-Sastre, R. J., Acevedo-Rodríguez, J., and Maldonado-Bascón, S. (2014). Recognizing in the depth: Selective 3d spatial pyramid matching kernel for object and scene categorization. *Image and Vision Computing*, 32(12):965 – 978. 6, 160
- [Ren et al., 2015a] Ren, S., He, K., Girshick, R., and Sun, J. (2015a). Faster r-cnn: Towards real-time object detection with region proposal networks. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems 28*, pages 91–99. Curran Associates, Inc. 126, 131
- [Ren et al., 2015b] Ren, S., He, K., Girshick, R., and Sun, J. (2015b). Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NIPS)*. 131
- [Rituerto et al., 2014] Rituerto, A., Murillo, A., and Guerrero, J. (2014). Semantic labeling for indoor topological mapping using a wearable catadioptric system. *Robotics and Autonomous Systems*, 62(5):685 – 695. Special Issue Semantic Perception, Mapping and Exploration. 106
- [Romero and Cazorla, 2010] Romero, A. and Cazorla, M. (2010). *Topological SLAM Using Omnidirectional Images: Merging Feature Detec-*

- tors and Graph-Matching*, pages 464–475. Springer Berlin Heidelberg, Berlin, Heidelberg. 109
- [Rumelhart et al., 1986] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088):533–536. 19
- [Russakovsky et al., 2015] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252. 12, 21, 23, 24, 27, 83, 127, 131, 167, 177, 179, 180, 184
- [Rusu et al., 2009] Rusu, R., Blodow, N., and Beetz, M. (2009). Fast point feature histograms (fpfh) for 3d registration. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 3212–3217. 41, 42
- [Se et al., 2005] Se, S., Lowe, D., and Little, J. (2005). Vision-based global localization and mapping for mobile robots. *IEEE Transactions on Robotics*, 21(3):364–375. 106
- [Sermanet et al., 2013] Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., and LeCun, Y. (2013). Overfeat: Integrated recognition, localization and detection using convolutional networks. *CoRR*, abs/1312.6229. 126, 131
- [Sharif Razavian et al., 2014] Sharif Razavian, A., Azizpour, H., Sullivan, J., and Carlsson, S. (2014). Cnn features off-the-shelf: An astounding baseline for recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. 107
- [Shi et al., 2015] Shi, B., Bai, S., Zhou, Z., and Bai, X. (2015). Deeppano: Deep panoramic representation for 3-d shape recognition. *IEEE Signal Processing Letters*, 22(12):2339–2343. 129
- [Sinha et al., 2016] Sinha, A., Bai, J., and Ramani, K. (2016). *Deep Learning 3D Shape Surfaces Using Geometry Images*, pages 223–240. Springer International Publishing, Cham. 130

- [Sipiran and Bustos, 2011] Sipiran, I. and Bustos, B. (2011). Harris 3d: a robust extension of the harris operator for interest point detection on 3d meshes. *The Visual Computer*, 27(11):963–976. 40
- [Sivic and Zisserman, 2003] Sivic, J. and Zisserman, A. (2003). Video Google: A text retrieval approach to object matching in videos. In *International Conference on Computer Vision*, pages 1470–1477, Nice, France. IEEE. 87
- [Smith et al., 1987] Smith, R., Smith, R., Self, M., and Cheeseman, P. (1987). Estimating uncertain spatial relationships in robotics. In Self, M., editor, *Robotics and Automation. Proceedings. 1987 IEEE International Conference on*, volume 4, pages 850–850. 3, 157
- [Song et al., 2014] Song, H. O., Girshick, R., Jegelka, S., Mairal, J., Harchaoui, Z., and Darrell, T. (2014). On learning to localize objects with minimal supervision. In Jebara, T. and Xing, E. P., editors, *ICML - 31st International Conference on Machine Learning*, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 1611–1619, Beijing, China. JMLR. 132
- [Song et al., 2015] Song, S., Lichtenberg, S. P., and Xiao, J. (2015). Sun rgb-d: A rgb-d scene understanding benchmark suite. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 567–576. 5, 159
- [Srivastava et al., 2014] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958. 19, 175
- [Steinberg et al., 2015] Steinberg, D. M., Pizarro, O., and Williams, S. B. (2015). Hierarchical bayesian models for unsupervised scene understanding. *Computer Vision and Image Understanding*, 131:128 – 144. Special section: Large Scale Data-Driven Evaluation in Computer Vision. 5, 160

- [Stergiopoulou and Papamarkos, 2006] Stergiopoulou, E. and Papamarkos, N. (2006). A new technique for hand gesture recognition. In *Image Processing, 2006 IEEE International Conference on*, pages 2657–2660. 27, 183
- [Su et al., 2015] Su, H., Maji, S., Kalogerakis, E., and Learned-Miller, E. (2015). Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 945–953. 130
- [Sünderhauf et al., 2015a] Sünderhauf, N., Dayoub, F., Shirazi, S., Upcroft, B., and Milford, M. (2015a). On the performance of convnet features for place recognition. *CoRR*, abs/1501.04158. 107
- [Sünderhauf et al., 2015b] Sünderhauf, N., Shirazi, S., Jacobson, A., Dayoub, F., Pepperell, E., Upcroft, B., and Milford, M. (2015b). Place recognition with convnet landmarks: Viewpoint-robust, condition-robust, training-free. In *Robotics: Science and Systems*, Auditorium Antonianum, Rome. 107
- [Szegedy et al., 2014] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2014). Going deeper with convolutions. *CoRR*, abs/1409.4842. 20, 23, 114, 176, 179
- [Tang et al., 2016] Tang, Y., Wang, J., Gao, B., Dellandréa, E., Gaizauskas, R., and Chen, L. (2016). Large scale semi-supervised object detection using visual and semantic knowledge transfer. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2119–2128. 131
- [Thrun et al., 2002] Thrun, S. et al. (2002). Robotic mapping: A survey. *Exploring artificial intelligence in the new millennium*, pages 1–35. 82, 105
- [Thrun and Leonard, 2008] Thrun, S. and Leonard, J. (2008). Simultaneous localization and mapping. In *Springer handbook of robotics*, pages 871–889. Springer. 105

- [Tombari and Di Stefano, 2010] Tombari, F. and Di Stefano, L. (2010). Object recognition in 3d scenes with occlusions and clutter by hough voting. In *Image and Video Technology (PSIVT), 2010 Fourth Pacific-Rim Symposium on*, pages 349–355. 33, 128
- [Tombari et al., 2011] Tombari, F., Gori, F., and Di Stefano, L. (2011). Evaluation of stereo algorithms for 3d object recognition. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 990–997. 32, 126, 128
- [Tombari and Salti, 2011] Tombari, F. and Salti, S. (2011). A combined texture-shape descriptor for enhanced 3d feature matching. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pages 809 –812. 6, 41, 161
- [Tombari et al., 2010] Tombari, F., Salti, S., and Di Stefano, L. (2010). Unique signatures of histograms for local surface description. In *Proceedings of the 11th European conference on computer vision conference on Computer vision: Part III, ECCV’10*, pages 356–369, Berlin, Heidelberg. Springer-Verlag. 6, 161
- [Tombari et al., 2013] Tombari, F., Salti, S., and Di Stefano, L. (2013). Performance evaluation of 3d keypoint detectors. *International Journal of Computer Vision*, 102(1-3):198–220. 40
- [Tudhope and Taylor, 1997] Tudhope, D. and Taylor, C. (1997). Navigation via similarity: automatic linking based on semantic closeness. *Information Processing & Management*, 33(2):233–242. 84
- [Tung and Little, 2015] Tung, F. and Little, J. J. (2015). Improving scene attribute recognition using web-scale object detectors. *Computer Vision and Image Understanding*, 138:86 – 91. 6, 160
- [Tung and Little, 2016] Tung, F. and Little, J. J. (2016). Scene parsing by nonparametric label transfer of content-adaptive windows. *Computer Vision and Image Understanding*, 143:191 – 200. Inference and Learning of Graphical Models Theory and Applications in Computer Vision and Image Analysis. 5, 159

- [Uijlings et al., 2013] Uijlings, J. R. R., van de Sande, K. E. A., Gevers, T., and Smeulders, A. W. M. (2013). Selective search for object recognition. *International Journal of Computer Vision*, 104(2):154–171. 131
- [Vaca-Castano et al., 2017] Vaca-Castano, G., Das, S., Sousa, J. P., Lobo, N. D., and Shah, M. (2017). Improved scene identification and object detection on egocentric vision of daily activities. *Computer Vision and Image Understanding*, 156:92 – 103. Image and Video Understanding in Big Data. 6, 160
- [Valgren et al., 2006] Valgren, C., Lilienthal, A., and Duckett, T. (2006). Incremental topological mapping using omnidirectional vision. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 3441–3447. IEEE. 83, 84
- [Viejo et al., 2012] Viejo, D., Garcia, J., Cazorla, M., Gil, D., and Johnson, M. (2012). Using GNG to improve 3d feature extraction-application to 6dof egomotion. *Neural Networks*. 32
- [Wang et al., 2015] Wang, C., Huang, K., Ren, W., Zhang, J., and Maybank, S. (2015). Large-scale weakly supervised object localization via latent category learning. *IEEE Transactions on Image Processing*, 24(4):1371–1385. 132
- [Wang and Lin, 2011] Wang, M.-L. and Lin, H.-Y. (2011). An extended-ht semantic description for visual place recognition. *The International Journal of Robotics Research*, 30(11):1403–1420. 106
- [Whelan et al., 2016] Whelan, T., Salas-Moreno, R. F., Glocker, B., Davison, A. J., and Leutenegger, S. (2016). Elasticfusion: Real-time dense slam and light source estimation. *Intl. J. of Robotics Research, IJRR*. 109
- [Wohlkinger and Vincze, 2011] Wohlkinger, W. and Vincze, M. (2011). Ensemble of shape functions for 3d object classification. In *Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on*, pages 2987–2992. IEEE. 64, 138

- [Wu et al., 2012] Wu, D., Zhu, F., and Shao, L. (2012). One shot learning gesture recognition from rgb-d images. In *Conference on Computer Vision and Pattern Recognition Workshops*, pages 7–12. 108
- [Wu et al., 2009] Wu, J., Christensen, H., Rehg, J. M., et al. (2009). Visual place categorization: Problem, dataset, and algorithm. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4763–4770. IEEE. 60
- [Wu et al., 2016] Wu, J., Zhang, C., Xue, T., Freeman, W. T., and Tenenbaum, J. B. (2016). Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. *arXiv preprint arXiv:1610.07584*. 129
- [Wu et al., 2015] Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., and Xiao, J. (2015). 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1912–1920. 129
- [Xu et al., 2013] Xu, G., Mourrain, B., Duvigneau, R., and Galligo, A. (2013). Analysis-suitable volume parameterization of multi-block computational domain in isogeometric applications. *Computer-Aided Design*, 45(2):395 – 404. Solid and Physical Modeling 2012. 35
- [Zhong, 2009] Zhong, Y. (2009). Intrinsic shape signatures: A shape descriptor for 3d object recognition. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 689–696. 40
- [Zhou et al., 2014] Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., and Oliva, A. (2014). Learning deep features for scene recognition using places database. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 27*, pages 487–495. Curran Associates, Inc. 5, 12, 22, 23, 25, 26, 83, 85, 94, 130, 160, 167, 178, 179, 181, 183
- [Óscar Martínez Mozos et al., 2007] Óscar Martínez Mozos, Triebel, R., Jensfelt, P., Rottmann, A., and Burgard, W. (2007). Supervised se-

mantic labeling of places using information extracted from sensor data.  
*Robotics and Autonomous Systems*, 55(5):391 – 402. From Sensors to  
Human Spatial Concepts. 106



Universitat d'Alacant  
Universidad de Alicante





Universitat d'Alacant  
Universidad de Alicante

# List of Acronyms

---

**ANNs** *Artificial Neural Networks*

**API** *Application Programming Interface*

**BoVW** *Bag of Visual Words*

**BoW** *Bag of Words*

**Caffe** *Convolutional Architecture for Fast Feature Embedding*

**CAT-SLAM** *Continuous Appearance-Based Trajectory SLAM*

**CNN** *Convolutional Neural Networks*

**DL** *Deep Learning*

**ESF** *Esemble of Shape Functions*

**FLANN** *Fast Library for Approximate Nearest Neighbors*

**FPFH** *Fast Point Feature Histograms*

**GNG** *Growing Neural Gas*

**ICP** *Iterative Closest Point*

**IDOL** *Image Database for Robot Localization*

**ILSVRC** *ImageNet Large Scale Visual Recognition Challenge*

**ISS** *Intrinsic Shape Signatures*

**kNN** *k-Nearest Neighbors*

**K-S** *Kolmogórov-Smirnov Distance*

**NG** *Neural Gas*

**PCA** *Principal Component Analysis*

**PCL** *Point Cloud Library*

**PHOG** *Pyramid Histogram of Oriented Gradients*

**ReLU**s *Rectifier Linear Units*

**RF**s *Random Forests*

**RGB-D** *Image Format: Red Green Blue Depth*

**RTAB-MAP** *Real-Time Appearance-Based Mapping*

**SHOT** *Unique Signatures of Histograms for Local Surface Description*

**SLAM** *Simultaneous Localization And Mapping*

**SP** *Spin Image*

**SVM** *Support Vectors Machine*

**SVM-DAS** *Support Vectors Machine with Discriminative Accumulation Scheme*

**ViDRIO** *Visual and Depth Robot Indoor Localization with Objects information dataset*