UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

YGOR CÉSAR NOGUEIRA SOUSA

A New Approach to Semantic Mapping Using Reusable Consolidated Visual Representations

Recife
2023

YGOR CÉSAR NOGUEIRA SOUSA

A New Approach to Semantic Mapping Using Reusable Consolidated Visual Representations

A Ph.D. Thesis presented to the Centro de Informática of Universidade Federal de Pernambuco in partial fulfillment of the requirements for the degree of Philosophy Doctor in Computer Science.

Concentration Area: Computational Intelligence

Advisor: Hansenclever de França Bassani

Recife

2023

Catalogação na fonte
Bibliotecária Nataly Soares Leite Moro, CRB4-1722

S725n Sousa, Ygor César Nogueira
A new approach to semantic mapping using reusable consolidated visual representations / Ygor César Nogueira Sousa – 2023.
92 f.: il., fig., tab.

Orientador: Hansenclever de França Bassani.
Tese (Doutorado) – Universidade Federal de Pernambuco. CIn, Ciência da Computação, Recife, 2023.
Inclui referências.

1. Inteligência computacional. 2. Mapeamento semântico topológico. 3. Robótica móvel. I. Bassani, Hansenclever de França (orientador). II. Título

006.31 CDD (23. ed.) UFPE - CCEN 2024 – 17

Ygor César Nogueira Sousa

“A New Approach to Semantic Mapping Using Reusable Consolidated Visual Representations”

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Doutor em Ciência da Computação. Área de Concentração: Inteligência Computacional

Aprovada em: 28/08/2023.

_____

Orientador: Prof. Dr. Hansenclever de França Bassani

BANCA EXAMINADORA

_____

Profa. Dra. Edna Natividade da Silva Barros
Centro de Informática / UFPE

_____

Prof. Dr. Aluizio Fausto Ribeiro Araujo
Centro de Informática / UFPE

_____

Prof. Dr. Adiel Teixeira de Almeida Filho
Centro de Informática / UFPE

_____

Prof. Dr. Douglas Guimarães Macharet
Departamento de Ciência da Computação/UFMG

_____

Prof. Dr. João Fausto Lorenzato de Oliveira
Escola Politécnica de Pernambuco/ UPE

I dedicate this thesis to my family.

ACKNOWLEDGEMENTS

First of all, I want to thank God for giving me strength at all times and allowing me to overcome the adversities that were imposed on me during the development of this research.

I want to thank my family, especially my father, José Cícero, my mother, Angela, and my sister, Natália, for all their support, attention and patience during this journey. This achievement is not just mine, it belongs to all of us.

I also want to thank my advisor, Hansenclever Bassani, for his dedication, availability, patience and guidance over all these years. It was truly an honor to work with you.

Furthermore, I want to thank my colleagues in the laboratory and at UFPE, for the moments of fun, cooperation and learning. I also thank all the professors at UFPE who shared some of their knowledge with me throughout this journey.

Finally, I thank all my closest friends for their usual support and everyone who has supported me in some way over all these years.

ABSTRACT

The advancement of robotics may produce a positive impact on several aspects of our society. However, in order for robotic agents to assist humans in a variety of everyday activities, they need to possess representations of their environments that allow spatial and human-centered semantic understanding. Many works in the recent literature use Convolutional Neural Network (CNN) models to recognize semantic properties of images and incorporate the results into traditional metric or topological maps, a procedure known as semantic mapping. The types of semantic properties (e.g., room size, place category, and objects) and their semantic classes (e.g., kitchen and bedroom, for place category) are usually previously defined and restricted to the planned tasks. Thus, all the visual data acquired and processed during the construction of the maps is lost, and only the recognized semantic properties remain on the maps. In contrast, this research proposes using the visual data acquired during the mapping process to create reusable representations of regions by consolidating deep features extracted from the data. These consolidated representations would allow the recognition of new semantic information in a flexible way, and consequently, the adaptation of the semantics of the maps to new requirements of new tasks without the need for remapping. Such use of reusable consolidated representations for the generation of semantic maps is demonstrated in a topological mapping method that creates consolidated representations of deep visual features extracted from RGB images captured around each topological node. This is done using a process we denote as Topological Consolidation of Features by Moving Averages (TCMA). Experiments performed with real-world indoor datasets suggested that the proposed method is able to create consolidated representations that fairly preserve the visual features of the original images they consolidated and do not degrade in quality over time. Furthermore, the very promising results suggested that the consolidated representations produced are suitable for recognizing different semantic properties, indicating the topological location of images and adapting previously created maps with new semantic information. The experiments included two different CNNs for deep features extraction, classifiers trained on large-scale datasets from the literature, and more practical real-time scenarios. Different variations of the method were evaluated, including a derivation of the TCMA process that uses the arithmetic mean of multiple exponential moving averages.

Keywords: topological semantic mapping; deep learning for visual perception; representation learning; visual features consolidation; transfer learning; mobile robotics.

RESUMO

O avanço da robótica pode produzir um impacto positivo em vários aspectos da nossa sociedade. No entanto, para que os agentes robóticos auxiliem os seres humanos em uma variedade de atividades cotidianas, eles precisam possuir representações de seus ambientes que permitam a compreensão espacial e semântica centrada em seres humanos. Muitos trabalhos na literatura recente usam Redes Neurais Convolucionais (CNN, do inglês *Convolutional Neural Network*) para reconhecer propriedades semânticas de imagens e incorporam os resultados em mapas métricos ou topológicos tradicionais, um procedimento conhecido como mapeamento semântico. Os tipos de propriedades semânticas (ex: tamanho do cômodo, categoria de lugar e objetos) e suas classes semânticas (ex: cozinha e quarto, para categoria de lugar) geralmente são previamente definidos e restritos às tarefas planejadas. Assim, todos os dados visuais adquiridos e processados durante a construção dos mapas são perdidos, restando apenas as propriedades semânticas reconhecidas nos mapas. Em contraste, esta pesquisa propõe usar os dados visuais adquiridos durante o processo de mapeamento para criar representações reutilizáveis de regiões pela consolidação de características visuais profundas extraídas dos dados. Essas representações consolidadas permitiriam o reconhecimento de novas informações semânticas de forma flexível e, consequentemente, a adaptação da semântica dos mapas a novos requisitos de novas tarefas sem a necessidade de remapeamento. O uso de representações consolidadas reutilizáveis para a geração de mapas semânticos é demonstrado em um método de mapeamento topológico que cria representações consolidadas de características visuais profundas extraídas de imagens RGB capturadas em torno de cada nó topológico. Isso é feito usando um processo que denominamos como Consolidação Topológica de Características por Médias Móveis (TCMA, do inglês *Topological Consolidation of Features by Moving Averages*). Experimentos realizados com conjuntos de dados de ambientes internos do mundo real sugeriram que o método proposto é capaz de criar representações consolidadas que preservam as características visuais das imagens originais consolidadas e não degradam a qualidade ao longo do tempo. Além disso, os resultados promissores sugeriram que as representações consolidadas produzidas são adequadas para o reconhecimento de diferentes propriedades semânticas, a localização topológica de imagens e adaptação de mapas criados anteriormente com novas informações semânticas. Os experimentos incluíram duas CNNs diferentes para extração de características profundas, classificadores treinados em conjuntos de dados de larga escala da literatura e cenários mais práticos com execução em tempo real. Diferentes variações do método foram avaliadas, incluindo uma derivação do processo TCMA que usa a média aritmética de múltiplas médias móveis exponenciais.

Palavras-chave: mapeamento semântico topológico; aprendizagem profunda para percepção visual; aprendizagem de representações; consolidação de características visuais; aprendizagem por transferência; robótica móvel.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

CNN	Convolutional Neural Network
COCO	Common Objects in Context
COLD	COsy Localization Database
DSSOM	Dimension Selective Self-Organizing Map
HOG	Histogram of Oriented Gradients
LARFDSSOM	Local Adaptive Receptive Field Dimension Selective Self-Organizing Map
LARFSOM	Local Adaptive Receptive Field Self-Organizing Map
LHS	Latin Hypercube Sampling
MLP	Multilayer Perceptron
NBNN	Naive Bayes Nearest Neighbor
ReLU	Rectified Linear Unit
ROS	Robot Operating System
SLAM	Simultaneous Localization and Mapping
SOM	Self-Organizing Maps
SVM	Support Vector Machine
TCCMA	Topological Consolidation of Features by the Combination of Multiple Moving Averages
TCMA	Topological Consolidation of Features by Moving Averages

CONTENTS

1 INTRODUCTION

Recent advances in machine learning and robotics have resulted in a growing number of applications of artificial autonomous agents, such as robots, in people's everyday environments to perform many complex tasks in homes, industries, transportation, and so on. For these agents to be able to adequately assist humans in a variety of activities, they need to have representations of their environments that go beyond spatial understanding. The spatial representations, built as metric or topological maps from on-board sensors data, contain geometric information that allow robotic agents to navigate through environments while locating themselves, planning trajectories, and avoiding obstacles. However, spatial understanding alone does not allow the agents to differentiate the type of place where they are (e.g., living room or bedroom) or which objects are present in the room, that is, it does not allow them to understand the world through human concepts (ACHOUR et al., 2022).

Without human-centered understanding, it would be intractable for robotic agents to perform complex tasks common in everyday human life, such as responding to a natural language command (e.g., "clean all the bedrooms" or "go to the kitchen and get an apple for me"), interacting with the environment if necessary (e.g., opening a door to enter a destination or picking up a requested object), or avoiding unpleasant behavior (e.g., standing in front of a door for too long) (HAN et al., 2021). Therefore, understanding the world through human concepts is essential for robots to coexist with humans and assist them in everyday activities. To address this issue, a procedure known in the literature as semantic mapping is used to allow robotic agents to associate geometric entities in the spatial representations with human-centered semantic information (KOSTAVELIS; GASTERATOS, 2015).

Semantic mapping methods incorporate into traditional spatial maps (such as metric or topological) human-centered semantic properties that vary depending on the type of environments they represent. In indoor environments (such as offices or homes), for example, semantic properties like place categories (e.g., kitchen, corridor, and bathroom) and objects (e.g., chair, bed, and television) may be important (PRONOBIS; JENSFELT, 2012). In the literature, the semantic properties are generally obtained automatically from sensory data (such as RGB images and 2D depth measurements) using machine learning models. However, some approaches also infer semantic properties from other semantic information already obtained (e.g., objects recognized in a room may be used to infer the place category) or receive them through human input (e.g., semantic properties of critical contexts can be defined manually, such as road signs) (KOSTAVELIS; GASTERATOS, 2015; HAN et al., 2021; ACHOUR et al., 2022).

The recent development of deep learning, especially in Convolutional Neural Networks (CNN), allowed the automatic acquisition of semantic properties of the mapped environments at a new level. Currently, CNN models are the state-of-the-art for various computer vision tasks (GU et al., 2018; LI et al., 2022) and in semantic mapping they are widely used to recognize semantic properties from images through classification, objects detection, and semantic seg-

mentation (HAN et al., 2021; ACHOUR et al., 2022; CHEN et al., 2022). The common approach in semantic mapping is to recognize the semantic properties from individual images and fuse (or accumulate) them over time into the associated spatial elements, as information recognized in just one instant and view rarely provides a reliable semantic understanding (KOSTAVELIS; GASTERATOS, 2015).

However, training deep learning models like CNNs requires lots of data and is very costly in terms of time and computational resources. Because of this, many modern semantic mapping methods use CNNs already trained (on large-scale datasets) to solve a related problem. The pre-trained CNNs are used to perform recognition of semantic properties mainly in two ways: without any changes or retraining (SUNDERHAUF et al., 2017; SOUSA; BASSANI, 2018; RANGEL et al., 2019; ROSINOL et al., 2021); or they are customized through transfer learning to recognize task-specific semantic information (SUNDERHAUF et al., 2016; MCCORMAC et al., 2017; RODDICK; CIPOLLA, 2020).

Even with the powerful addition of deep learning models for the automatic recognition of semantic properties from images, modern semantic mapping methods still have the semantics incorporated in the maps restricted to the types of semantic properties defined previously to support the planned application scenarios. This occurs because the standard procedure in semantic mapping methods is to use the acquired visual data to recognize the semantic properties and associate them with the spatial entities only during the mapping process, that is, the visual data acquired during the mapping process is used to recognize the semantic properties and then discarded. Thus, no new semantic information can be recognized from the lost data and only the semantic labels are kept in the maps. This restricts the application of the produced maps to new tasks, which would require at least a remapping process with new semantic properties recognition models or the use of other mechanisms (like human input).

Differently, this research investigates how to create semantic maps that can automatically update their semantics using machine learning methods even after the mapping process and proposes that a promising path is to use the large amount of visual data acquired during the process to create visual representations of spatial regions that consolidate visual features extracted from multiple images. *Consolidation* in this research is defined as the process of combining several feature vectors into a single vector that is representative of the whole. Hence, a *consolidated visual representation* is the result of the consolidation of multiple visual representations (i.e., the visual features vectors) extracted from individual images by a latent layer of a pre-trained CNN model.

These consolidated visual representations could be reused for the recognition of certain semantic properties during the mapping process and would allow the recognition of new semantic information to adapt the maps to fulfill the requirements of new tasks defined in the future. Furthermore, all tasks that use path planning could automatically benefit from new semantic information added to the maps for more efficient path planning, which could only be accomplished by remapping all environments in the traditional procedure. Therefore, the use

of reusable consolidated visual representations could improve the applicability of the maps in several aspects.

Given the above, two objectives emerge for this thesis. The first is to develop a semantic mapping method with an innovative procedure of adaptive semantic association made from reusable consolidated visual representations and validate its efficacy through the use of the consolidated representations for the flexible recognition of different semantic properties. The second objective is to develop a process of deep visual features consolidation that can effectively preserve relevant visual features of the consolidated images and richly represent the visual characteristics of the associated spatial regions. The second objective aims to enable the first objective with a specific consolidation process integrated into the semantic mapping method and not to develop a general-purpose visual features consolidation method. Additionally, as indoor and outdoor environments face different application challenges, this thesis restricts its objectives to indoor environments.

To address the presented objectives, this thesis first proposes a topological semantic mapping method that creates consolidated representations of the deep visual features extracted from images captured around each topological node. The deep visual features are extracted using a pre-trained CNN and are consolidated through a process we denote as Topological Consolidation of Features by Moving Averages (TCMA). The consolidation process uses exponential moving averages, is inspired by Self-Organizing Maps (SOM) with time-varying structure (ARAUJO; REGO, 2013), and has visual persistence and visual habituation capabilities. Experiments performed in a real-world indoor dataset suggest that the consolidated visual representations produced are rich representations of the topological regions they cover, fairly preserve the visual features of the consolidated images, and can be used to recognize varied semantic properties (such as place category, room, and objects), as well as to indicate the topological location of images (a distinct visual task).

Then, this thesis introduces a version of the topological semantic mapping method for real-time operation and a derivation of the TCMA process that combines multiple exponential moving averages, the TCCMA (Topological Consolidation of Features by the Combination of Multiple Moving Averages) process. The real-time version of the method is evaluated with both proposed consolidation processes, separately, in experiments that aimed to assess the creation and application of consolidated visual representations in more practical scenarios. The results suggest that the consolidated visual representations produced are suitable for the accurate recognition of different semantic properties using classifiers trained on large-scale datasets found in the literature and can be reused to adapt the semantic information of previously created maps.

Achieving the presented objectives is not an easy task, but the proposed solutions proved to be very promising, which may represent a starting point for a new perspective for the generation of semantic maps.

1.1 THESIS OUTLINE

The rest of this thesis is organized as follows.

- Chapter 2 presents important concepts that contributed to the development of the solutions proposed in this research;
- Chapter 3 presents a brief review of the literature on semantic mapping and contextualizes the solutions proposed in this research;
- Chapter 4 details the proposed topological semantic mapping method and the experiments performed to evaluate the maps produced with it. In the experiments, the quality and applicability of the consolidated visual representations are evaluated;
- Chapter 5 details the version of the proposed method for real-time operation and the TCCMA process. The chapter also details and discusses the experiments performed to evaluate the application of the method and its variations in more practical scenarios;
- Chapter 6 discusses the contributions of this research, describes limitations identified in the proposed solutions and proposes directions for future work.

2 BACKGROUND

This chapter presents important concepts that contributed to the development of the solutions proposed in this thesis. First, Section 2.1 provides a brief introduction to deep learning, CNNs, and transfer learning, and also briefly describes GoogLeNet (SZEGEDY et al., 2015) and MobileNetV2 (SANDLER et al., 2018), the two CNN models used in the proposed solutions. Then, Section 2.2 details the main concepts of SOM and describes LARFDSSOM (Local Adaptive Receptive Field Dimension Selective Self-Organizing Map) (BASSANI; ARAUJO, 2015), a self-organizing map with time-varying structure. Finally, Section 2.3 discusses the presented concepts that inspired certain characteristics of the solutions proposed in this thesis.

2.1 DEEP LEARNING

For decades, considerable domain expertise was required to design feature extractors, such as SIFT (LOWE, 1999) and SURF (BAY et al., 2008), that transformed raw data (e.g: the pixels of an image) into adequate feature vectors (also called representations) from which machine learning methods could detect or classify patterns (LECUN; BENGIO; HINTON, 2015). However, the performance of machine learning methods is strongly dependent on the data representation on which they are applied and such domain expertise feature extractors are unable to extract discriminate information from the data itself, in addition to requiring careful human engineering (BENGIO; COURVILLE; VINCENT, 2013).

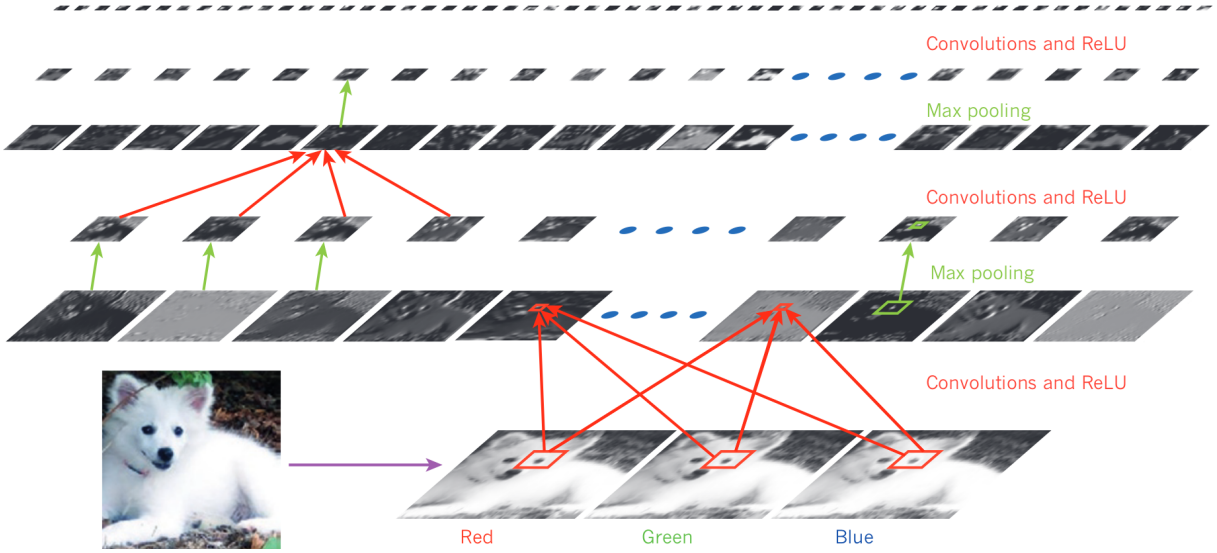
For this reason, much effort has been made in the literature to allow machine learning methods to automatically discover from raw data the necessary representations for tasks such as detection and classification. Deep learning methods are representation learning techniques that learn multiple levels of representations by the composition of multiple non-linear modules. Starting with the raw input, each module transforms the representation of the data at one level into a slightly more abstract representation level. Very complex patterns can be learned by composing several of these transformations (LECUN; BENGIO; HINTON, 2015).

One of the most prominent deep learning approaches is the Convolutional Neural Network (CNN). Currently, CNN-based models are the dominant approach in many computer vision tasks (such as face recognition, object detection and semantic segmentation) and have allowed the research community to make significant advances in very complex applications (such as autonomous vehicles and medical diagnostics) (LECUN; BENGIO; HINTON, 2015; GU et al., 2018; LI et al., 2022). The next section briefly describes the main components of a CNN.

2.1.1 Convolutional Neural Networks

CNN is a feedforward neural network designed to process input data in grid-like form. An RGB image, for example, is a 3D grid composed of three 2D arrays containing pixel intensities

Figure 1 – The outputs of convolutional and pooling layers in a typical CNN architecture applied to an RGB image. Each rectangular image is a feature map.



Source: Adapted from LeCun, Bengio and Hinton (2015).

(one for each color channel) (GOODFELLOW; BENGIO; COURVILLE, 2016). The basic architecture of a CNN is a sequence of several layers and each layer transforms the input volume (a 3D matrix in the case of an RGB image) into a 3D output volume (also called representation). The main three types of layers in a CNN are: convolutional, pooling and fully connected.

The role of the convolutional layer is to detect local conjunctions of features from the previous layer (LECUN; BENGIO; HINTON, 2015). A convolutional layer is composed of a set of learnable filters and each of them produces a different 2D feature map that makes up the 3D output volume of the layer. Each filter has a different kernel (i.e.: a matrix of weights) for each channel in the previous layer. The number of channels in the previous layer is determined by the number of feature maps produced in that layer, or, if it is the first layer, by the number of channels in the input data (e.g.: 3, for an RGB image). All kernels in a filter have the same size (frequently called of filter size), usually $N \times N$ with N small (such as 5×5 or 3×3), and all filters in a traditional convolutional layer have the same filter size.

Each kernel is applied to all sub-regions $N \times N$ of its respective channel in the input volume. For each sub-region, it is computed the sum of the results of the element-wise multiplication between the weights in the kernel and the values in the sub-region. Therefore, each kernel produces a processed version of its respective channel and all processed versions are summed to produce the output 2D feature map. If a layer has n filters, n 2D feature maps are produced and concatenated to form the 3D output volume for the next layer. Additionally, an activation function, such as ReLU (Rectified Linear Unit) or sigmoid, is usually applied to each element in the resulting 3D output volume before passing it to the next layer. In a CNN architecture, the first convolutional layer learns to detect low-level features (such as curves and edges) and subsequent layers learn to detect higher-level features representations from the output of the previous layers, gradually.

The pooling layer acts to reduce the dimensions of the input feature maps and create invariance for small shifts and distortions (LECUN; BENGIO; HINTON, 2015). A pooling layer is commonly inserted between convolutional layers and resizes each 2D feature map in the input volume, separately (i.e.: the number of channels in the output volume remains unchanged). Each element in a resized 2D feature map is typically the result of maximum or average operation of all elements in a $N \times N$ sub-region in the input representation. Figure 1 exemplifies the operation of convolutional and pooling layers in an RGB image.

In a CNN, after a sequence of several convolutional and pooling layers, it is common to add fully connected layers (one or more). A fully connected layer in a CNN works in the same way as in a regular MLP (Multilayer Perceptron), and therefore has multiple units, each connected to all units in the input volume by a set of learnable weights, and produces a 2D output vector. Some CNN architectures use convolutional layers as final layers, so the use of fully connected layers is not obligatory (GU et al., 2018). The output layer in a CNN depends on the application. In a multiclass classification task, the output layer is usually a fully connected layer with the number of units equal to the number of classes selected, followed by a softmax function, which transforms the input vector into a vector of probabilities that sum to 100%.

Training the weights of a CNN is similar to training a regular deep neural network, by backpropagating the gradient of the error through the layers of the network (LECUN; BENGIO; HINTON, 2015). The error is calculated between the prediction outputs and the expected values in the training data using a loss function, such as the mean squared error and cross entropy (often used for regression and classification tasks, respectively) (LI et al., 2022).

Typically, if the dataset used to train the CNN is very large and general, the CNN learns to detect general features and its inner layers can be transferred to a second CNN, which will reuse the learning for a new target task. This procedure is defined as transfer learning, which allows the reuse of knowledge from one domain to another, helps to increase the generalization power of the trained models and to decrease the training time (YOSINSKI et al., 2014; NEYSHABUR; SEDGHI; ZHANG, 2020). The next section discusses transfer learning in more detail.

2.1.2 Transfer Learning

In the usual transfer learning process, the first n layers of a pre-trained CNN are transferred to a new CNN which uses them as its first n layers, then the subsequent layers of the new CNN are added and the network is trained on the data of the new target task. Training is usually done in one of two ways: the weights of the transferred layers are left frozen, which means that they will not change during training and the transferred layers will act as a features extractor; or, the errors obtained during training are backpropagated through all layers of the CNN, that is, the weights of the transferred layers will be used as initialization and finetuned to the new task (YOSINSKI et al., 2014).

The choice of which procedure to use depends on several factors, two of them are: the

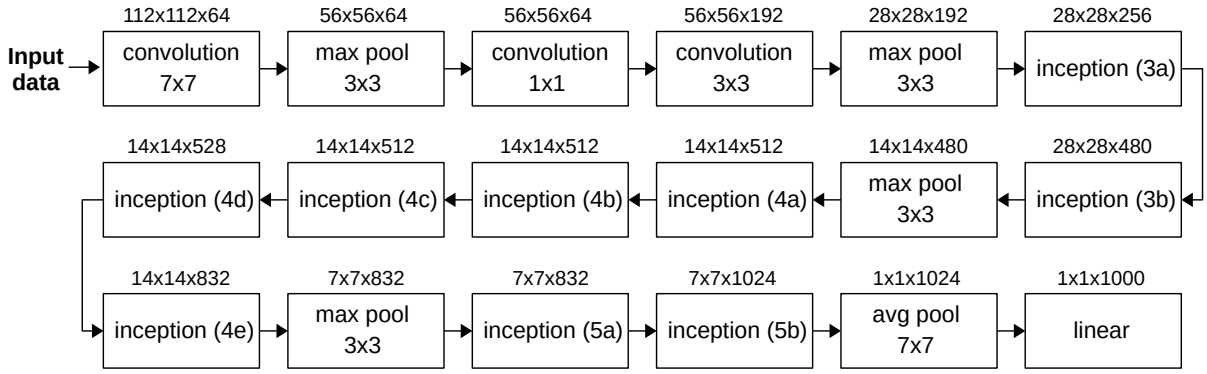
size of the target dataset and the number of layers (more precisely, the number of weights) transferred. For example, if the number of weights is small or the target dataset is large, the transferred weights can be finetuned to improve the performance in the new task. However, if the number of weights transferred is large and the target dataset is small, it is preferred to freeze the transferred layers to avoid overfitting. In this way, transfer learning allows the training of very effective deep networks with small datasets. (YOSINSKI et al., 2014).

The use of a CNN trained on a very large (and general) dataset as a fixed features extractor to transfer learning to a task of interest is a widely used procedure nowadays and is also used in the solutions proposed in this thesis. The next sections briefly describe GoogLeNet and MobileNetV2, the two CNN models used in this thesis. The models are used pre-trained on ImageNet-1K, a subset of ImageNet (DENG et al., 2009) with 1000 image categories. ImageNet[1] is a large-scale image database organized by the hierarchical structure of WordNet (FELLBAUM, 1998) with over 14 million annotated images and more than 20.000 categories.

2.1.3 GoogLeNet

GoogLeNet is the convolutional neural network that achieved the best classification results in the ImageNet Large-Scale Visual Recognition Challenge 2014 (ILSVRC14) (SZEGEDY et al., 2015). The challenge involved classifying images among the 1000 classes of ImageNet-1K, which contains approximately 1.2 million images for training, 50.000 for validation and 100.000 for testing. GoogLeNet starts with some traditional convolutional and pooling layers, and then stacks several Inception modules (with the occasional use of pooling layers in between). Its basic architecture is illustrated in Figure 2.

Figure 2 – GoogLeNet basic architecture. Each block is a layer or Inception module. The number below a block label is the size of the filters and the number above a block is the output size. For example: in the first convolutional layer, the filters have size 7x7 and the output size of the layer is 112x112x64.



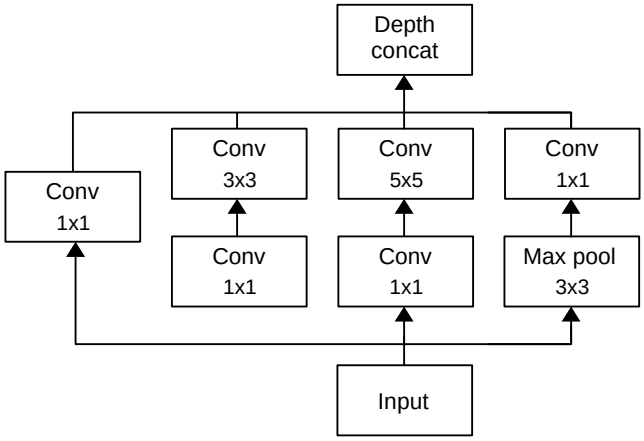
Source: Adapted from Szegedy et al. (2015).

An Inception module (illustrated in Figure 3) applies convolution filters of three different sizes (1×1, 3×3, and 5×5) in parallel, and the resulting feature maps are concatenated

[1] URL: https://www.image-net.org

to produce the output of the module. This design allows the layer after the module to receive as input feature maps extracted at different scales concatenated into a single representation. Furthermore, the 1×1 convolution filters are used to reduce the number of channels before the 3×3 and 5×5 convolution filters, which allowed the CNN to increase its depth and width without significant performance penalty. ReLU activation functions are applied after all convolution filters in the architecture, including those within Inception modules. We refer the reader to the original paper (SZEGEDY et al., 2015) for further details on the model and its architecture.

Figure 3 – An Inception module from GoogLeNet.



Source: Adapted from Szegedy et al. (2015).

2.1.4 MobileNetV2

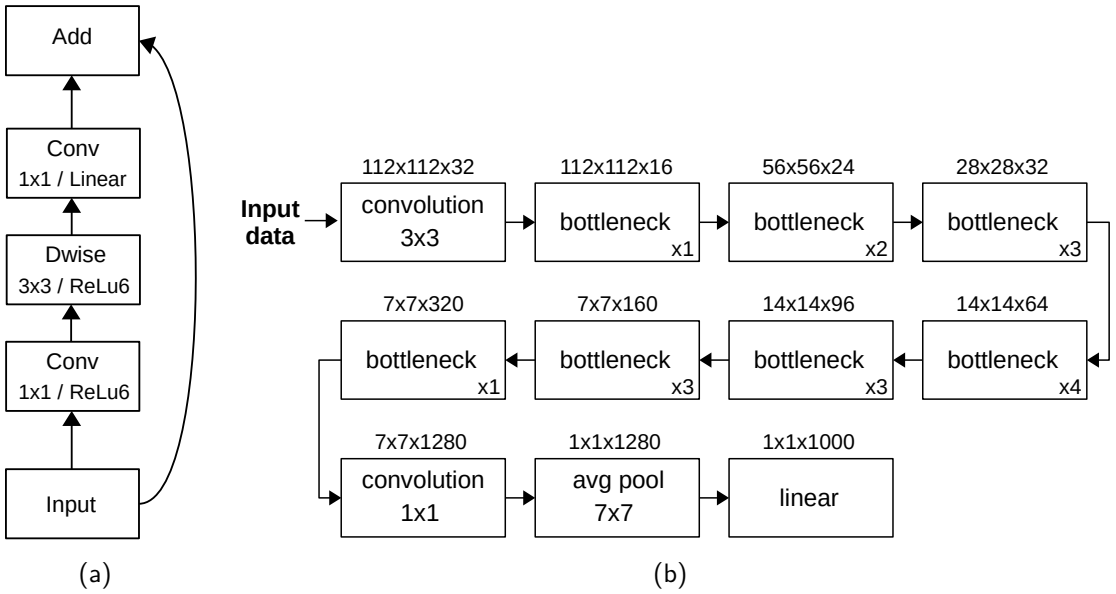
MobileNetV2 (SANDLER et al., 2018) is a convolutional neural network designed for mobile and resource-constrained environments, which, while maintaining accuracy, significantly decreases the number of operations and memory used. The model is based on MobileNetV1 (HOWARD et al., 2017), a CNN that uses Depthwise Separable Convolutions as its main building block. In a traditional convolutional layer, each filter is applied to all channels in the input representation. Differently, a Depthwise Separable Convolution block divides the convolutional process into two layers: the first is the depthwise convolution layer, which applies a single convolutional filter per input channel; and the second is a convolutional layer with 1×1 filters (also called pointwise convolution layer), which calculates linear combinations of the output channels of the depthwise convolution layer. This design significantly reduces the computational cost when compared to traditional convolutional layers.

MobileNetV2 extends the first version by introducing a module (illustrated in Figure 4), called Inverted Residual with Linear Bottleneck, which first expands the number of channels in the input representation using a convolutional layer with 1×1 filters, then applies a depthwise convolution layer with 3×3 filters, then a linear convolutional layer with 1×1 filters (also

called linear bottleneck layer in this architecture) projects the resulting features back to a representation with the original number of channels, and the obtained representation is added to the original input representation through a shortcut connection. The use of such shortcut connections allows the model to increase its ability to propagate gradients across layers. Furthermore, ReLU6 activation functions are applied after each convolutional layer in the module, with the exception of the linear convolutional layer, since (as the authors suggest) using non-linearities at bottlenecks can destroy too much information and degrade performance. ReLU6 limits the positive direction of the function to a maximum value and was used because it showed robustness to low-precision computations.

There are two version of the module, the first is as described and the second does not use the shortcut connection, i.e., the output of the module is that of the linear convolutional layer (the last convolutional layer of the module). The architecture of the CNN starts with a traditional convolutional layer followed by several Inverted Residual with Linear Bottleneck modules, as illustrated in Figure 4.

Figure 4 – In (a) an Inverted Residual with Linear Bottleneck module from MobileNetV2 is illustrated, where Dwise stands for depthwise convolution layer. In (b) the basic architecture of MobileNetV2 is presented, where each block is a layer or Bottleneck module (Bottleneck stands for Inverted Residual with Linear Bottleneck module). The number below a block label is the size of the filters and the number above a block is the output size. If the block is a Bottleneck module, the number in the bottom right corner is the number of repetitions of the module and the number above a block is the size of the output after all repetitions.



Source: Adapted from Sandler et al. (2018).

For further details on the MobileNetV2 and its architecture, we refer the reader to the original paper (SANDLER et al., 2018). For more general details on deep learning and CNNs, we refer the reader to (LECUN; BENGIO; HINTON, 2015; GOODFELLOW; BENGIO; COURVILLE, 2016), on representation learning to (BENGIO; COURVILLE; VINCENT, 2013), and transfer learning for

deep models to (YOSINSKI et al., 2014; NEYSHABUR; SEDGHI; ZHANG, 2020). Also, for recent reviews on CNNs, we refer the reader to (GU et al., 2018; LI et al., 2022).

2.2 SELF-ORGANIZING MAPS

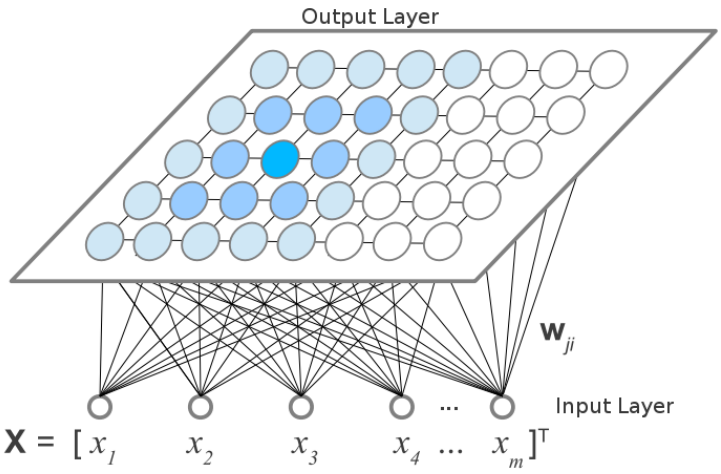
Self-Organizing Maps (SOM) is a neural network originally proposed by Kohonen (1982) as one of the first neural networks with unsupervised learning between the 1970s and 1980s, and typically maps a high-dimensional distribution to a smaller regular grid in a compression process that preserves the topological and metric relationships of the input data. SOM was originally proposed for high-dimensional data visualization, but it is also frequently used for other problems such as quantization and data clustering (HAYKIN, 1998). The next section describes the main components of the SOM of Kohonen.

2.2.1 SOM of Kohonen

The basic structure of a SOM is formed by an input layer and an output layer. The input layer propagates the input data to all nodes in the output layer, a regular and usually two-dimensional grid of nodes, where a topological structure is formed with the output nodes summarizing the input data received over time.

The input layer is a vector $\mathbf{x} = \{x_i, i = 1...m\}$, where m is the number of dimensions of the input data. The output layer is a grid of nodes, where each node j contains a vector of weights $\mathbf{w}_j = \{w_{ji}, i = 1...m\}$ and is connected to its immediate neighbors. Thus, each node in the output layer has a weight for each dimension in the input vector. Figure 5 illustrates the mentioned structure.

Figure 5 – Basic structure of the SOM of Kohonen in rectangular grid.



Source: Adapted from Bassani (2014).

The SOM learning process consists of three steps: competition, where the nodes in the

output layer compete and the node most similar to the input data is the winner; adaptation, where the winning node is updated to better represent the input data; and cooperation, where the neighbors of the winning nodes are also updated.

In this way, whenever an input pattern $\mathbf{x}$ is presented to the map, there is a competition and the winner is the one with the smallest distance between the weights vector $\mathbf{w}_j$ and the input pattern $\mathbf{x}$:

$$b(\mathbf{x}) = \arg\min_j [D(\mathbf{x}, \mathbf{w}_j)], \tag{2.1}$$

where the distance $D(\mathbf{x}, \mathbf{w}_j)$ is usually calculated as the Euclidean distance, as follows:

$$D(\mathbf{x}, \mathbf{w}_j) = \sqrt{\sum_{i=1}^{m} (x_i - w_{ji})^2}. \tag{2.2}$$

Then, the winning node has its weights vector $\mathbf{w}_j$ adapted in direction of the input data $\mathbf{x}$:

$$\mathbf{w}_j(n+1) = \mathbf{w}_j(n) + \eta(n) h_{jb(x)}(n)(\mathbf{x} - \mathbf{w}_j(n)), \tag{2.3}$$

where $\eta(n)$ is a learning rate calculated by a function that decays with the learning iterations:

$$\eta(n) = \eta_0 exp\left(- \frac{n}{\tau_1} \right), \tag{2.4}$$

where τ_1 is a parameter that adjusts the decay speed.

The term $h_{jb(x)}$ is the neighborhood decay function, a function used in the adaptation of the winning node and in the cooperation step, where the neighbor nodes of the winning node are also updated in direction of the input data $\mathbf{x}$ but with a reduced level of adaptation usually determined by a Gaussian, as follows:

$$h_{jb(x)} = exp\left(\frac{|| \mathbf{r}_j - \mathbf{r}_b ||^2}{2\sigma^2(n)} \right), \tag{2.5}$$

where $\mathbf{r}_b$ and $\mathbf{r}_j$ are the positions of the winning node b and the neighboring node j in the output layer grid. The function $\sigma(n)$ starts with value σ_0 and decreases with the training iterations:

$$\sigma(n) = \sigma_0 exp\left(- \frac{n}{\tau_2} \right), \tag{2.6}$$

where τ_2 is a parameter that adjusts the decay speed.

The output layer in the basic structure of the SOM of Kohonen has a fixed and previously defined number of nodes and connections between them. Thus, prior knowledge of the data is necessary to obtain good results. A way to minimize this need is found in the literature, where some works introduce models with topological structures (number of nodes and connections) that can vary dynamically over time (FRITZKE, 1994; KUNZE; STEFFENS, 1995; MARSLAND; SHAPIRO; NEHMZOW, 2002; ARAUJO; REGO, 2013). LARFDSSOM is one such model, in which, in addition to having a time-varying topological structure, it also learns a relevance value for each dimension in the input data. This is different from what is done in the SOM of Kohonen,

where all dimensions of the input data are considered equally by the distance metric. Employing the same weight for all dimensions may cause problems in the clustering process, especially in high-dimensional data, as some dimensions may be relevant to a cluster and irrelevant to another (BASSANI, 2014).

Some concepts from LARFDSSOM and SOM in general, inspired characteristics present in the solutions proposed in this thesis. Therefore, the next section describes the operation of LARFDSSOM in more detail.

2.2.2 LARFDSSOM

LARFDSSOM (Local Adaptive Receptive Field Dimension Selective Self-Organizing Map) is a SOM with time-varying topological structure proposed by (BASSANI; ARAUJO, 2015), which is based on two previous methods: DSSOM (Dimension Selective Self-Organizing Map) (BASSANI; ARAUJO, 2012), a SOM that applies different relevance to each input dimension in the nodes of the output layer; and, on LARFSOM (Local Adaptive Receptive Field Self-Organizing Map) (ARAUJO; COSTA, 2009), a SOM that modifies its topological structure when necessary.

The LARFDSSOM learning process has the same steps of competition, adaptation, and cooperation present in the SOM of Kohonen and is divided into two phases: the self-organization phase, and the convergence phase. In the self-organization phase, the three steps are repeated for several epochs, the nodes that do not reach a minimum winning percentage are removed from the map, and whenever no node on the map reaches a minimum activation value for the input pattern, a new node is added to the map. Then, in the convergence phase, the self-organization process continues without adding new nodes, the nodes that do not reach a minimum winning percentage are removed from the map, and the process continues until the number of nodes on the map stops decreasing.

The next sections detail the LARFDSSOM learning process through the steps of competition, adaptation and cooperation.

2.2.2.1 Competition

In LARFDSSOM, each node j on the map (output layer) contains three vectors: the center vector $\mathbf{c}_j = \{c_{ji}, i = 1...m\}$, which is the weight vector of the SOM of Kohonen renamed to center of the cluster and m is the number of dimensions in the input data; the relevance vector $\boldsymbol{\omega}_j = \{\omega_{ji}, i = 1...m\}$, in which each component represents a weight, within $[0, 1]$, for each input dimension; and the average distance vector $\boldsymbol{\delta}_j = \{\delta_{ji}, i = 1...m\}$, which is a moving average of the distance between the input data and the center vector.

For each input pattern $\mathbf{x}$, the winner of a competition is the node on the map with the

highest activation $ac(D_\omega(\mathbf{x}, \mathbf{c}_j), \boldsymbol{\omega}_j)$.

$$s(\mathbf{x}) = \arg\max_j [ac(D_\omega(\mathbf{x}, \mathbf{c}_j), \boldsymbol{\omega}_j)]. \tag{2.7}$$

The activation $ac(D_\omega(\mathbf{x}, \mathbf{c}_j), \boldsymbol{\omega}_j)$ of a node j is calculated by a radial basis function of the weighted distance $D_\omega(\mathbf{x}, \mathbf{c}_j)$ and the norm of the relevance vector $|| \boldsymbol{\omega}_j ||$, as follows:

$$ac(D_\omega(\mathbf{x}, \mathbf{c}_j), \boldsymbol{\omega}_j) = \frac{1}{1 + \frac{D_\omega(\mathbf{x}, \mathbf{c}_j)}{(|| \boldsymbol{\omega}_j ||^2 + \varepsilon)}}, \tag{2.8}$$

where ε is a small value to avoid division by zero and $D_\omega(\mathbf{x}, \mathbf{c}_j)$ is the distance between the input pattern $\mathbf{x}$ and the center vector $\mathbf{c}_j$ of the node j, weighted by the relevance vector, calculated as follows:

$$D_\omega(\mathbf{x}, \mathbf{c}_j) = \sqrt{\sum_{i=1}^{m} \omega_{ij}(x_i - c_{ji})^2}. \tag{2.9}$$

If the winning node has an activation lower than the activation threshold a_t, no node is adapted and a new node η is added to the map, where $\mathbf{c}_\eta = \mathbf{x}$, $\boldsymbol{\omega}_\eta$ has all its components initialized with value one, and $\boldsymbol{\delta}_\eta$ has all its components initialized with value zero. Otherwise, the winning node and its neighbors are adapted.

2.2.2.2 Adaptation and Cooperation

First, the center vector $\mathbf{c}_j$ of the winning node and its neighbors are adapted in direction to the input data by the moving average below, using two learning rates, $e_b \in]0, 1[$ for the winning node and $e_n \in]0, e_b[$ for the neighbors.

$$\mathbf{c}_j(n+1) = \mathbf{c}_j(n) + e(\mathbf{x} - \mathbf{c}_j(n)), \tag{2.10}$$

where $e = e_b$ if j is the winning node or $e = e_n$ if j is one of its neighbors.

Next, the winning node and its neighbors have their average distance vectors $\boldsymbol{\delta}_j$ updated through a moving average of the distance between the center of the node and the input pattern:

$$\boldsymbol{\delta}_j(n+1) = (1 - e\beta)\boldsymbol{\delta}_j(n) + e\beta(|\mathbf{x} - \mathbf{c}_j(n)|), \tag{2.11}$$

where $|\mathbf{x} - \mathbf{c}_j(n)|$ is the absolute value of the difference between $\mathbf{x}$ and $\mathbf{c}_j$, $\beta \in]0, 1[$ controls the rate of change of the moving average, and $e = e_b$ if j is the winning node or $e = e_n$ if j is one of its neighbors.

Then, each component of the relevance vector $\boldsymbol{\omega}_j$ of the winning node and its neighbors is updated through an inverse logistic function of the average distance:

$$\omega_{ji} = \begin{cases} \dfrac{1}{1 + exp\left(\dfrac{\delta_{ji} - \delta_{jimean}}{s(\delta_{jimax} - \delta_{jimin})}\right)}, & \text{if } \delta_{jimin} \neq \delta_{jimax} \\ 1, & \text{otherwise} \end{cases} \tag{2.12}$$

where δ_{jimin}, δ_{jimax} and δ_{jimean} are the minimum value, the maximum value and the average of the components of the distance vector $\boldsymbol{\delta}_j$. The parameter $s > 0$ controls the slope of the function.

Each node j on the map has the variable $cwins_j$ that counts its number of wins in competitions. Whenever a defined number of competitions occurs, nodes that have not reached the minimum winning percentage lp are removed from the map and the number of wins for all remaining nodes on the map is reset to zero. Furthermore, to prevent removal of nodes recently added, when a new node j is added, its number of wins $cwins_j$ is initialized with the value $lp \times ncomps$, where $ncomps$ is the number of competitions since the last removal.

After a node removal occurs, the connections between all the remaining nodes are updated. In addition, when a new node is added to the map, the connections between the new node and all others are updated. LARFDSSOM uses the similarity between the relevance vectors of two nodes to determine if there is a connection between them, so a connection between nodes means that they cluster similar sub-spaces in the input data. The equation below details the connectivity criterion between two nodes:

$$\text{nodes i and j are } \begin{cases} \text{connected}, & \text{if } \|\boldsymbol{\omega}_i - \boldsymbol{\omega}_j\| < c\sqrt{m} \\ \text{disconnected}, & \text{otherwise} \end{cases} \tag{2.13}$$

where $\|\boldsymbol{\omega}_i - \boldsymbol{\omega}_j\|$ is the norm of the difference between the relevance vectors, m is the number of dimensions in the input data, and c is the connection threshold usually set to 0.5, which means that two nodes with $\|\boldsymbol{\omega}_i - \boldsymbol{\omega}_j\|$ below half the maximum $\sqrt{m}$ are connected.

For further details on the LARFDSSOM, we refer the reader to the original paper (BASSANI; ARAUJO, 2015). For more information about other SOM models with time-varying structure, we refer the reader to (ARAUJO; COSTA, 2009; ARAUJO; REGO, 2013), and about the original SOM to (KOHONEN, 1982; KOHONEN, 1990).

2.3 INSPIRATIONS

The concept of transfer learning in deep neural networks inspired the use in the proposed method of deep representations extracted from individual images, by a CNN pre-trained on a large-scale (and more general) dataset, for the creation of consolidated representations of multiple images that could be reused for other tasks related to, but different from the one for which the CNN used was originally trained. Furthermore, transfer learning was a fundamental aspect to enable the training of classifiers to recognize semantic properties from these consolidated representations, since they are trained using the same pre-trained CNN used to extract the original representations. An important observation is that at the beginning of this research, we did not know if the classifiers would be able to accurately recognize semantic properties from the consolidated representations or if the consolidated representations would be able to maintain the main characteristics of the individual images.

Regarding self-organizing maps, the concept of time-varying topological structure of the LARFDSSOM inspired the way the topological maps generated with the proposed method add new spatial nodes and update the winning nodes, however, in the proposed methods the competitions are made using the Euclidean distance as in the SOM of Kohonen. The competitions in the proposed method are employed to determine the topological location of the agent from received spatial coordinates, thus, the traditional Euclidean distance is adequate for the task and it is not necessary to apply weights or activation function.

In addition, the procedure for updating the consolidated vectors in the two proposed consolidation processes (TCMA and TCCMA) was inspired by the adaptation step of the SOM learning process, which uses an exponential moving average controlled by a learning rate to adapt its nodes and better represent the input data. In addition, in the TCCMA process, the concept of time-varying structure inspired how the consolidation items are created to represent different patterns in the data consolidated in each spatial node.

Finally, the topological image localization approach, introduced in Chapter 4 to demonstrate the application of the consolidated visual representations in a visual task other than the recognition of semantic properties, was based on LARFDSSOM. In the approach, the data consolidated in the nodes of the generated topological maps are used in competitions, where relevance vectors are applied to the dimensions of the data in the nodes and the winning nodes represent the location of the images on the maps.

3 SEMANTIC MAPPING

This chapter presents a brief review (i.e., a non-exhaustive review focused on the main characteristics) of the literature on semantic mapping and contextualizes the solutions proposed in this thesis. First, Section 3.1 discusses how semantic mapping is defined in the literature and describes its main characteristics. Then, Section 3.2 briefly defines the two types of maps (metric and topological) traditionally used in semantic mapping methods to represent the environments spatially. Next, Section 3.3 describes how the semantic properties incorporated into the maps are generally defined and acquired. Finally, Section 3.4 relates the solutions proposed in this thesis with the current literature. Section 3.4 is based on the contextualization with the literature made in our last published paper (SOUSA; BASSANI, 2022).

3.1 DEFINITION

Semantic mapping is defined by Pronobis (2011) as the process of building representations of environments that associate human concepts with instances of spatial entities. The result should ideally be an efficient and complete representation of the environment that contains not only semantic information, but also the spatial entities with which the semantics are associated. Following the same idea, Bastianelli et al. (2013) define semantic mapping as the process that allows robots to enrich maps used for navigation with semantic information about the environments. Duvallet (2015) adds that robots with models of their environments centered on humans may reason about high-level properties when interacting with people.

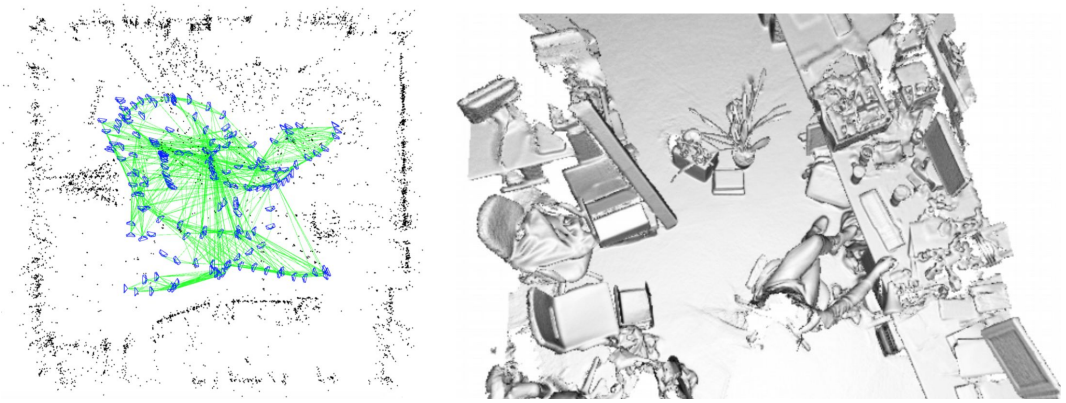
Han et al. (2021) indicate that there are three common characteristics in semantic mapping: First, a geometric map. It contains geometric information about the environment and is the base for attaching semantic information; Second, the semantic information. A human-understandable description of the environment that links physical entities to conceptual elements and bridges the semantic gap between robots and humans; Third, the geometric map and semantic information must have an organized structure that endows the robot with reasoning ability. Furthermore, Kostavelis and Gasteratos (2015) add that usually semantic maps are large-scale, i.e., maps that are created incrementally and have a global coordinate system.

Given the above, typical semantic mapping approaches take sensory data (such as RGB images, 2D or 3D depth measurements) as input, create spatial representations of the environment incrementally using traditional metric or topological mapping methods, and simultaneously annotate the spatial representations during the mapping process with human-centered semantic information (also called semantic properties) from the environment, such as objects and categories of place. The semantic information can also be obtained later, usually by inference or human input.

3.2 SPATIAL REPRESENTATIONS

Metric maps are two-dimensional or three-dimensional spatial representations that capture geometric properties of physical environments. The metric maps have a higher resolution of the environment when compared to topological maps, but come with a higher computational cost (THRUN, 2002). There are typically two types of metric representations: sparse and dense (examples in Figure 6). Traditionally, sparse maps are used for localization purposes, since they are feature-based and computationally more efficient than dense maps. However, classic sparse maps usually do not provide enough geometric information for autonomous navigation and dense maps are preferred for this task as they provide more detailed representations of the environment (NIETO; GUIVANT; NEBOT, 2006).

Figure 6 – Examples of metric maps: (a) shows a feature-based sparse 3D metric map of a room; and (b) shows a dense 3D metric map of a laboratory.



(a) (b)

Source: (a) - Mur-Artal, Montiel and Tardos (2015), Cadena et al. (2016); (b) - Newcombe et al. (2011).

In most semantic mapping approaches, metric maps are obtained using SLAM (Simultaneous Localization and Mapping) methods, which allow localization and mapping while moving in unknown environments (HAN et al., 2021; CHEN et al., 2022). SLAM consists of concurrently estimating the state (e.g., position and orientation) of the robotic agent equipped with onboard sensors and building the spatial representation (the map) of the environment that the sensors are perceiving (CADENA et al., 2016).

In SLAM, two-dimensional metric maps are usually modeled in two paradigms: occupancy-grid maps, which discretizes the environment in cells and assigns a probability of occupation to each cell (example in Figure 7); and landmark-based maps, which models the environment as a sparse set of landmarks (CADENA et al., 2016). The creation of three-dimensional geometric representations of the environments is more complicated and less standardized in the literature than the 2D case. Three-dimensional metric maps are modeled in different types of representations, one of the most used is the feature-based sparse representation, in which the environment is represented as a set of sparse 3D landmarks corresponding to discriminative

features detected of the scene (such as lines and corners) (CADENA et al., 2016).

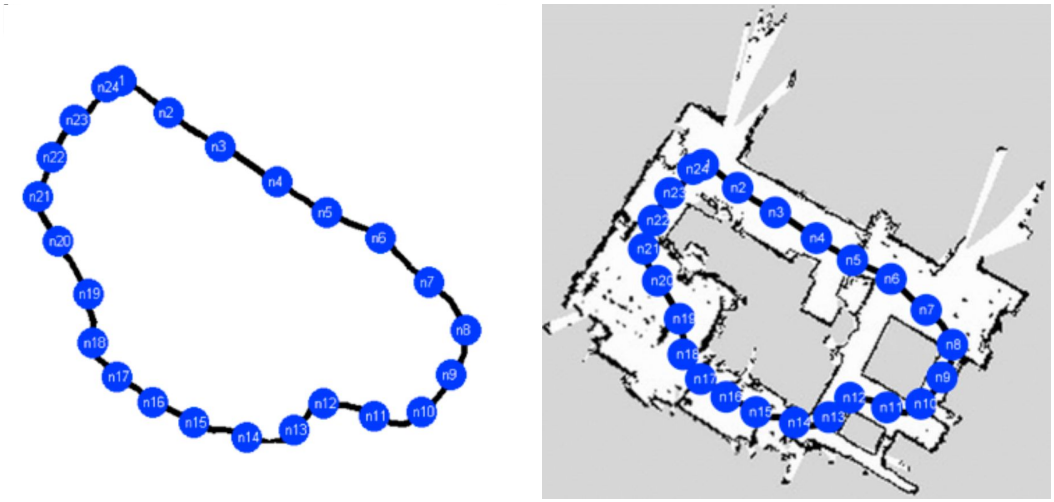
A SLAM method largely used in the semantic mapping literature that creates feature-based sparse 3D representations is ORB-SLAM2 (MUR-ARTAL; TARDOS, 2017), which is a version of the ORB-SLAM (MUR-ARTAL; MONTIEL; TARDOS, 2015) that supports monocular, stereo and RGB-D cameras. ORB-SLAM is a real-time monocular visual SLAM method that uses ORB descriptors (RUBLEE et al., 2011) for feature matching and is composed of three main threads: tracking, local mapping, and loop closing. Both methods can operate in small and large indoor and outdoor environments, and are widely used due to their robustness and real-time CPU performance (CHEN et al., 2022). ORB-SLAM2 is also used in this thesis to compose the integrated solution introduced in Chapter 5.

Another popular way of modeling three-dimensional metric maps is through low-level dense representations, which seek to model the geometry of the environment through high-resolution 3D representations usually described by a large set of points (i.e., point clouds) or polygons, or encoded as a set of disks (i.e., surfel maps). Furthermore, 3D metric maps are also often modeled to explicitly represent volumes in the environment through spatial-partitioning representations. A popular procedure is to discretize the space into 3D cubes (also called voxels) of fixed size and arrange them in a 3D grid. Additionally, many volumetric solutions model voxels in octrees for more optimized performance (CADENA et al., 2016).

On the other hand, topological maps offer more abstract representations of the environment (example in Figure 7). A topological map is typically created as a graph $G = (\mathbf{V}, \mathbf{E})$, in which $\mathbf{V} = \{v_j, j = 1...m\}$ is a vector of nodes where each node represents a distinct region in the environment, and $\mathbf{E} = \{e_i, i = 1...n\}$ is a vector of edges where each edge represents a transitivity relationship between two nodes. Two nodes are connected by an edge only if it is possible to move between them without going through any other node (ANATI, 2016).

As topological maps only store key spatial information, they consume less computational resources than metric maps and, therefore, are a good option for mapping large environments and for global path planning in the navigation process (ZHANG et al., 2023). However, topological maps are generally not suitable for local path planning as they lack detailed geometric information, being metric maps better suited to this type of task. Nodes and edges in topological maps may contain various types of spatial information, such as the spatial position of nodes or the metric distance between nodes. In semantic mapping methods, a common approach is to create topological maps on top of the metric ones (PRONOBIS; JENSFELT, 2012; HEMACHANDRA et al., 2014; LUO; CHIOU, 2018; ROSINOL et al., 2021) and the real-time version of the topological method proposed in this thesis follows this approach.

Figure 7 – An example of a 2D topological map of an indoor environment: (a) shows the topological map separately; and (b) shows the topological map on top of the associated 2D occupancy grid map. Each node in the topological map represents a distinct region of the occupancy grid.



(a) (b)

Source: Adapted from Kostavelis and Gasteratos (2015).

3.3 SEMANTIC ACQUISITION

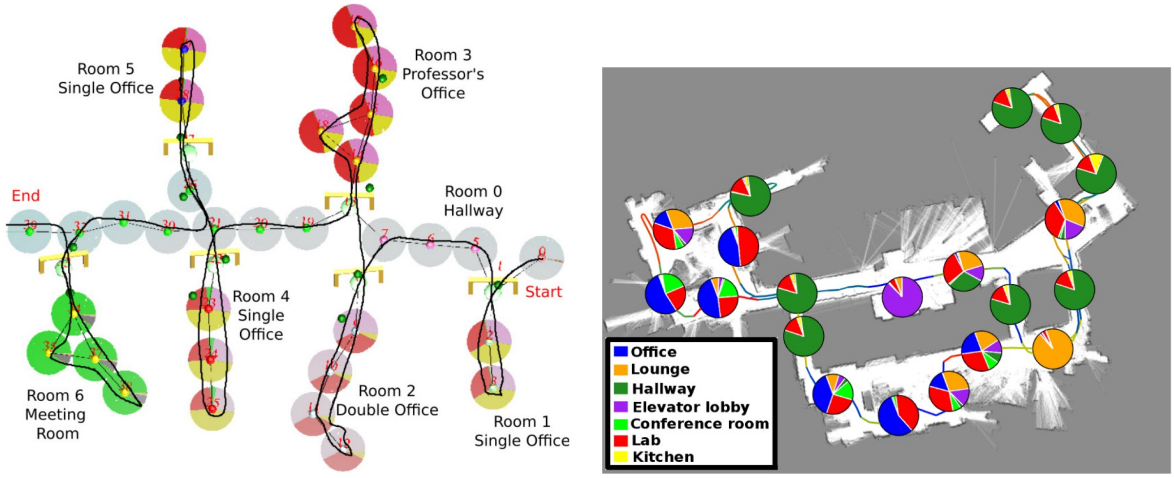
Typically, in semantic mapping techniques, the types of semantic properties (e.g., place category, objects, and room size) and their semantic classes (e.g., bathroom and kitchen, for place category) are defined in advance, and the semantic properties are automatically recognized using machine learning methods from the input sensory data acquired during the mapping process. Many recent semantic mapping approaches recognize the semantic properties through semantic segmentation, object detection, and classification using models based on CNNs (SUNDERHAUF et al., 2016; MATURANA; ARORA; SCHERER, 2017; LUO; CHIOU, 2018; GRINVALD et al., 2019; RODDICK; CIPOLLA, 2020; ROSINOL et al., 2021; CHEN et al., 2022).

Furthermore, Pronobis and Jensfelt (2012) point out that information acquired in just one instant and view of the environment rarely provides sufficient evidence for reliable semantic recognition and, therefore, any semantic mapping technique should integrate space and time in its building process. This concept is defined by Kostavelis and Gasteratos (2015) as temporal coherence, which means enriching maps with semantic properties acquired at different time instants.

Therefore, the traditional approach in the literature is to accumulate (or fuse) the semantic information recognized from different perspectives over time in the geometric units of the maps. In topological maps, the semantic properties are usually recognized by classification or object detection, and incorporated to the nodes or vertices of the maps (examples in Figure 8) (PRONOBIS; JENSFELT, 2012; HEMACHANDRA et al., 2014; BERNUY; SOLAR, 2018; SOUSA; BASSANI, 2018). In metric maps, the semantic properties are usually recognized from image

segmentation or object detection, and incorporated into the metric units or segments that compose the type of generated map (examples in Figure 9) (MCCORMAC et al., 2017; MA et al., 2017; MATURANA et al., 2018; NAKAJIMA et al., 2018; RODDICK; CIPOLLA, 2020; PAZ et al., 2020; ROSINOL et al., 2021).

Figure 8 – Examples of topological maps enriched with semantic information. In (a) the nodes contain probability distributions of room categories and in (b) of region categories. The probability distributions are represented by pie charts.



(a) (b)

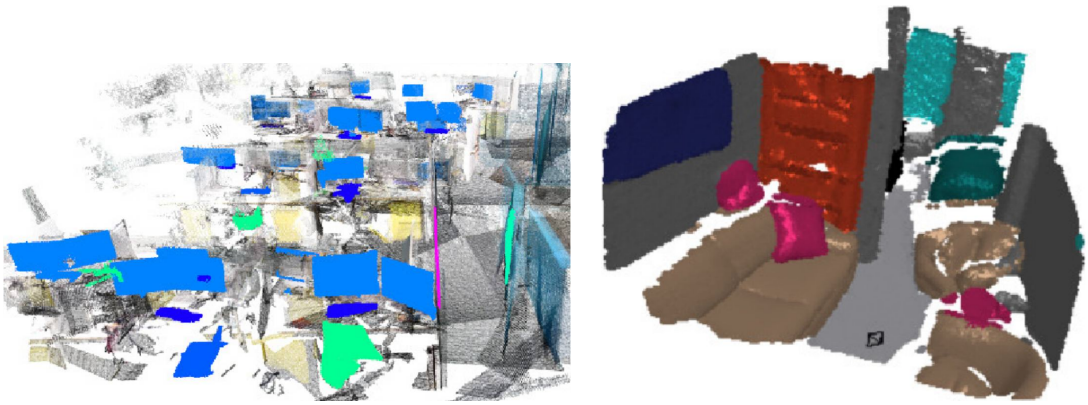
Source: (a) - Pronobis and Jensfelt (2012); (b) - Hemachandra et al. (2014).

In addition to the automatic acquisition of semantic properties through sensory data acquired during the mapping process, there are two other common ways of acquiring semantic properties in the literature (HAN et al., 2021; ACHOUR et al., 2022). The first is through interaction with humans. This procedure allows the inclusion of semantic information that would be difficult to automatically recognize, is not restricted to previous definitions (such as a set of place categories) and could be used even after the mapping process, however, it is heavily dependent on humans. Frequently, the semantic mapping approaches in the literature that acquire semantic information through human input use multimodal interfaces to allow interaction in different ways, such as through natural language, where humans can provide descriptions of places in natural language and enrich the maps with new semantic information (BASTIANELLI et al., 2013; KOLLAR et al., 2013; HEMACHANDRA et al., 2014; WALTER et al., 2014; DUVALLET, 2015; OH et al., 2015).

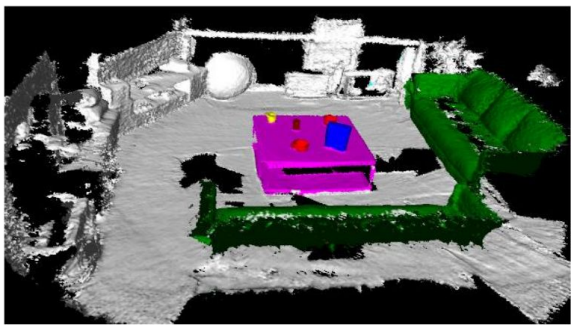
The second is through inference of additional semantic information from the semantic properties already obtained using the other procedures. In general, a knowledge database that contains knowledge common to humans is used by a reasoning engine to infer the additional information (ACHOUR et al., 2022). A typical example is the semantic mapping approach introduced by Pronobis and Jensfelt (2012), which uses a probabilistic chain graph (LAURITZEN; RICHARDSON, 2002) and a ontology of commonsense knowledge to infer the place categories of topological nodes from other semantic properties recognized using SVM (Support Vector

Machine) classifiers. Additionally, there are semantic mapping approaches in the literature that obtain additional semantic information from other semantic properties through clustering techniques, where the clusters represent the same semantic category (SOUSA; BASSANI, 2018; RANGEL et al., 2019).

Figure 9 – Examples of metric maps enriched with semantic information (each color represents a semantic class): (a) feature-based sparse 3D metric map of a lab environment; (b) surfel-based dense 3D metric map of a home environment; and (c) voxel-based dense 3D metric map of a home environment.



(a) (b)



(c)

Source: (a) - Sunderhauf et al. (2017); (b) - Nakajima et al. (2018); (c) - Xiang and Fox (2017).

3.4 RELATED WORK

Recently, CNNs have been widely used in the semantic mapping literature for recognizing semantic properties automatically from images in various types of approaches and applications (MA et al., 2017; NAKAJIMA et al., 2018; GRINVALD et al., 2019; PAZ et al., 2020; RODDICK; CIPOLLA, 2020). In (MCCORMAC et al., 2017), the authors use a CNN to perform semantic segmentation of images and incorporate the predictions into a dense 3D metric map. In a method created for Micro-Aerial Vehicles (MAV), Maturana, Arora and Scherer (2017) present a CNN designed for fast on-board processing. The CNN performs semantic segmentation of images and the 2D measurements are aggregated into a 2.5D grid map. In another work, Maturana et al. (2018) propose a CNN designed to perform semantic segmentation and incorporate the

recognized semantic information into a 2.5D grid map created for off-road autonomous driving. Bernuy and Solar (2018) use a CNN to perform semantic segmentation of images and the output labels are accumulated in histograms that are used to assist the creation of topological maps.

In a hybrid metric-topological semantic mapping approach, Luo and Chiou (2018) use a CNN to detect objects in a mapping system with multiple levels of semantic information. In (SUNDERHAUF et al., 2017), the authors use a pre-trained CNN to detect objects in images and integrate the results into a 3D sparse metric map, where an unsupervised 3D segmentation method is used to assign segments of 3D points of the map to the detected objects. Nakajima and Saito (2019) use a pre-trained CNN to detect objects in images and enhance 3D maps with object-oriented semantic information in an efficient framework. Rosinol et al. (2021) use a pre-trained CNN to perform semantic segmentation of images and incorporate the resulting labels into a 3D graph representation (the 3D Dynamic Scene Graph) that includes metric and semantic data from dynamic environments in different abstraction layers.

However, as mentioned earlier, the common approach in semantic mapping is to define in advance the types of semantic properties (and their classes) that will be recognized in the mapping process. They are predefined to attend the planned application scenarios, and the aforementioned methods do the same. Distinctly, some works use pre-trained CNNs integrated with other machine learning approaches to allow incremental learning of new classes of a semantic property. In (SUNDERHAUF et al., 2016), the authors introduce a model that incrementally learns new place categories, in a supervised fashion, by extending a CNN with an one-vs-all Random Forest classifier. Rangel et al. (RANGEL et al., 2019) present a semantic mapping method that exploits previously trained CNNs to classify unlabeled images and perform a bottom-up aggregation approach that clusters images in the same semantic category. Furthermore, in a previous work (SOUSA; BASSANI, 2018), we introduced a topological semantic mapping approach that incrementally and in an online fashion, forms clusters of object vectors classified with a pre-trained CNN. The clusters are formed using a SOM-based method and represent the categories of places visited. Still, even with the efforts to allow the semantic mapping methods to incorporate new semantic classes incrementally, these methods remain restricted to the previously defined types of semantic properties.

In addition, in the literature, the semantic properties are typically recognized by the CNNs from individual images and accumulated in the maps over time. However, despite the accumulation of information over time, this type of approach is limited only to what is seen locally in the image frame and loses the residual context of previous images that could improve the recognition accuracy. Differently, the method introduced by Xiang and Fox (2017) presents a recurrent approach that accumulates features extracted with a CNN to perform multi-view semantic segmentation in a joint 3D scene semantic mapping process. However, despite advances in the method that allow the accumulation of features from multiple views, it does not create representations of regions that can be used to obtain other semantic information and

the accumulated features are only used to label pixels of images. Furthermore, as with the other methods, the semantic labels are restricted to what was planned.

In contrast, this thesis introduces variations of a topological semantic mapping method that builds consolidated representations of deep visual features (i.e., extracted using a pre-trained CNN) from the regions delimited by the topological nodes, and the desired semantic information can be obtained afterward from these representations using a wide variety of machine learning methods. This presents an advantage over previous methods, as the consolidated representations are rich and allow new semantic information to be obtained whenever necessary, increasing the flexibility and applicability of the maps produced. To the best of our knowledge, this is the first approach in the semantic mapping literature with such characteristics, which even allow the adaptation (e.g., modification or enrichment) of the semantic information from previously created maps. The next chapter presents the proposed approach in detail.

4 TOPOLOGICAL SEMANTIC MAPPING USING CONSOLIDATED VISUAL REPRESENTATIONS

This chapter presents the topological semantic mapping method developed to address the objectives of this research. The method creates topologically consolidated representations of deep visual features, extracted from individual RGB images, using a process we denote as Topological Consolidation of Features by Moving Averages (TCMA). First, Section 4.1 describes the method and its consolidation process in detail. Then, Section 4.2 describes the experiments performed with the main objective of evaluating the quality and applicability of the consolidated representations of visual features produced. Most of this chapter is based on our last published paper (SOUSA; BASSANI, 2022).

4.1 THE METHOD

The topological semantic mapping method extracts deep visual features from RGB images and incrementally creates consolidated representations of the visual features extracted from the images captured around each topological node. The features are extracted from a deep latent layer of a pre-trained CNN and consolidated into the nodes of the topological map using the TCMA process. The CNN used is the GoogLeNet (SZEGEDY et al., 2015), pre-trained on ImageNet-1K and available in the PyTorch library (PASZKE et al., 2019).

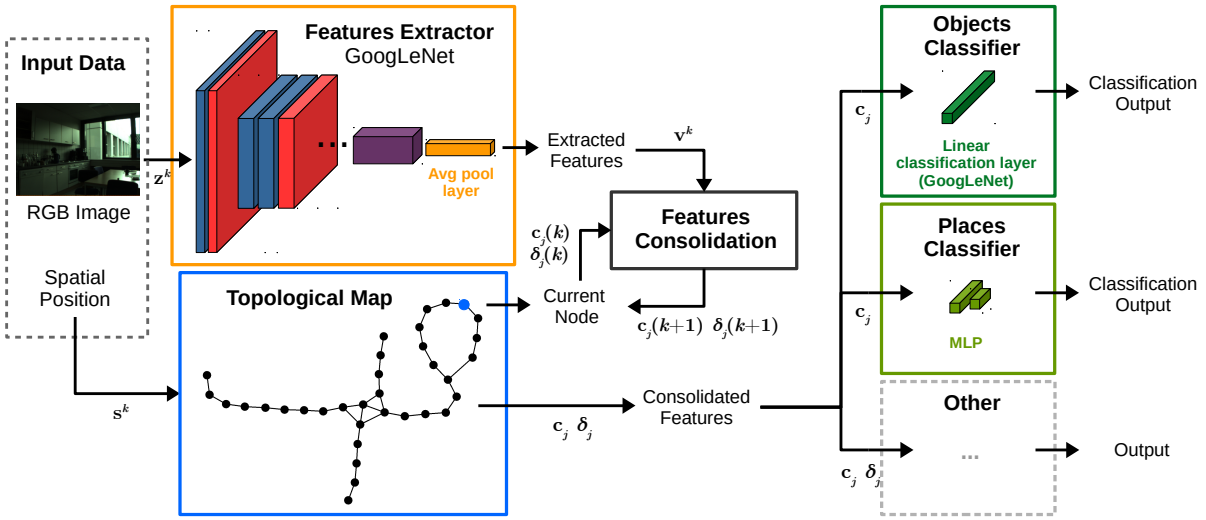
The TCMA process (detailed in Sections 4.1.1 and 4.1.2) uses exponential moving averages to perform the consolidations, is inspired by SOM with time-varying structure and, in addition, has two mechanisms inspired by the following neuroscience phenomena: visual habituation, which is described as the decrease in visual attention to repeated stimuli (FANTZ, 1964; COLOMBO; MITCHELL, 2009); and visual persistence, which refers to an optical illusion where a visual stimulus continues to be experienced for some time after its off-set (COLTHEART, 1980).

The consolidated representations are used to perform the classification of objects and place categories as semantic properties of the regions covered by the nodes. The objects are classified using the classification layer of GoogLeNet, without retraining, and the place categories are recognized using a shallow MLP (Multilayer Perceptron). The consolidated representations are also used to indicate the topological location of images with an approach also introduced in this chapter (Section 4.1.4) and could be used for the flexible recognition of other semantic properties. An overview of the method is illustrated in Figure 10 and the following sections describe its operation in detail.

4.1.1 Topological Mapping

At each time step k, the method receives as input an RGB image $\mathbf{z}^k$ and its spatial position of capture $\mathbf{s}^k = \{s_i^k, i = 1...n\}$ in the environment, where $n = 2$ in this scenario, but it can easily be extended to 3 dimensions for 3D maps (as demonstrated in Chapter 5). Each image

Figure 10 – Overview of the method: GoogLeNet extracts the visual features vector $\mathbf{v}^k$ of the input images and the nodes on the topological map corresponding to the input positions consolidate $\mathbf{v}^k$ in their representations; The consolidated visual features vector $\mathbf{c}_j$ in each node is provided to the classification layer of the GoogLeNet (without retraining) and the objects visualized in the nodes are obtained; The vector $\mathbf{c}_j$ in each node is also provided to a shallow MLP and the place category of each node is obtained; The consolidated vectors $\mathbf{c}_j$ and $\boldsymbol{\delta}_j$, the average features distance vector, could be used to recognize other semantic properties or in other visual tasks, as in the topological image localization experiments (Section 4.2.7) which uses $\mathbf{c}_j$ and $\boldsymbol{\delta}_j$.



Source: Thesis author.

is provided as input to the GoogLeNet, the output of the 2D adaptive average pooling layer (input of the linear classification layer) is flattened in the vector $\mathbf{v}^k$ of $m = 1024$ dimensions and used as a visual features vector.

The topological map starts empty and is built as a graph. Each node j in the graph contains three vectors: $\mathbf{p}_j = \{p_{ji}, i = 1...n\}$, the spatial position of the node; $\mathbf{c}_j = \{c_{ji}, i = 1...m\}$, the consolidated visual features vector; and $\boldsymbol{\delta}_j = \{\delta_{ji}, i = 1...m\}$, the average features distance vector.

As the input data ($\mathbf{v}^k$ and $\mathbf{s}^k$) is provided, the nodes compete to determine the topological location of the agent on the map and establish the nodes that will consolidate the visual features $\mathbf{v}^k$. The competition (ARAUJO; REGO, 2013) is performed using the spatial location provided, $\mathbf{s}^k$, and the winner is the node with the smallest spatial distance $D_e(\mathbf{s}^k, \mathbf{p}_j)$ to $\mathbf{s}^k$:

$$h(\mathbf{s}^k) = \arg\min_j [D_e(\mathbf{s}^k, \mathbf{p}_j)], \tag{4.1}$$

where $D_e(\mathbf{s}^k, \mathbf{p}_j)$ is calculated as the Euclidean distance between the given spatial position $\mathbf{s}^k$ and the spatial position of the node $\mathbf{p}_j$:

$$D_e(\mathbf{s}^k, \mathbf{p}_j) = \sqrt{\sum_{i=1}^{n} (s_i^k - p_{ji})^2}. \tag{4.2}$$

If the map is empty or the spatial distance of the winning node to $\mathbf{s}^k$ is greater than the spatial distance threshold λ, a new node η is inserted into the map with $\mathbf{p}_\eta = \mathbf{s}^k$, $\boldsymbol{\delta}_\eta = \mathbf{0}$,

and the consolidated visual features vector $\mathbf{c}_\eta$ initialized through an average between the input visual features $\mathbf{v}^k$ and the current state of the consolidated visual features vector $\mathbf{c}_l$ of the last visited node l, if any:

$$\mathbf{c}_\eta = \begin{cases} \gamma \mathbf{c}_l + (1-\gamma)\mathbf{v}^k, & \text{if there is a } \mathbf{c}_l \\ \mathbf{v}^k, & \text{otherwise,} \end{cases} \tag{4.3}$$

where γ is the persistence rate, which determines how much of the consolidated visual features vector of the last visited node will persist on the new node.

If the spatial distance of the winning node is equal to or smaller than the spatial distance threshold λ, the node consolidates the input visual features $\mathbf{v}^k$ as described in the next section (Section 4.1.2). Moreover, if there are other nodes with spatial distance higher, but equal to or smaller than λ, these nodes also consolidate the input visual features $\mathbf{v}^k$. This is because an image may be captured by the agent at an intersection position between nodes.

A connection between nodes is created whenever a transition occurs between them as the agent moves through the environment, and a transition is determined when the consecutive winning nodes are different. In addition, when a new node is inserted into the map, a connection is created between the new node and the previous winner, if any.

4.1.2 Visual Features Consolidation

The visual features in the input data are consolidated in the node j by updating the vector $\mathbf{c}_j$ through an exponential moving average considering the learning rate $\alpha \in]0,1[$ and the squared Euclidean distance, $D(\mathbf{v}^k, \mathbf{u}_j)$, between the visual features input vector $\mathbf{v}^k$ and the last visual features vector consolidated in node j, $\mathbf{u}_j$:

$$\mathbf{c}_j(k+1) = \begin{cases} \mathbf{c}_j(k) + \alpha(\mathbf{v}^k - \mathbf{c}_j(k)), & \text{if } D(\mathbf{v}^k, \mathbf{u}_j) \geq \tau \\ \mathbf{c}_j(k), & \text{otherwise,} \end{cases} \tag{4.4}$$

where τ is the minimum distance between the input features vector and the last features vector consolidated in the node, which prevents the node from consolidating very similar visual features vectors in sequence and, therefore, controls the visual habituation of the node. $D(\mathbf{v}^k, \mathbf{u}_j)$ is the squared Euclidean distance described as follows:

$$D(\mathbf{v}^k, \mathbf{u}_j) = \sum_{i=1}^{m} (v_i^k - u_{ji})^2. \tag{4.5}$$

Also through a moving average (introduced by Bassani and Araujo (2015)), the average features distance vector $\boldsymbol{\delta}_j$ of node j is updated considering the distance between $\mathbf{v}^k$ and $\mathbf{c}_j$:

$$\boldsymbol{\delta}_j(k+1) = \begin{cases} \boldsymbol{\delta}_j(k) + \alpha\beta(\boldsymbol{\phi} - \boldsymbol{\delta}_j(k)), & \text{if } D(\mathbf{v}^k, \mathbf{u}_j) \geq \tau \\ \boldsymbol{\delta}_j(k), & \text{otherwise,} \end{cases} \tag{4.6}$$

where $\beta \in]0,1[$ controls the rate of change of the moving average, α is the learning rate, $\boldsymbol{\phi} = |\mathbf{v}^k - \mathbf{c}_j(k)|$ denotes the absolute value of each component of the resulting difference vector, and τ controls the visual habituation of the node. After the update, if $\mathbf{c}_j$ and $\boldsymbol{\delta}_j$ were updated in relation to $\mathbf{v}^k$, then $\mathbf{u}_j = \mathbf{v}^k$.

4.1.3 Object and Place Classification

The visual features consolidated in the vector $\mathbf{c}_j$ are used to classify the objects visualized by the agent in the regions covered by each node j. For that, the vector $\mathbf{c}_j$ of each node is provided to the linear classification layer of GoogLeNet and the classification outputs are obtained without the need of retraining.

In addition, the visual features consolidated in the vector $\mathbf{c}_j$ are also used to obtain the place categories of the regions covered by each node j. A simple MLP implemented with the PyTorch library classifies the vector $\mathbf{c}_j$ of each node and the place category is obtained. The MLP contains one hidden layer with 20 units and ReLU activation functions. Furthermore, the loss function used for training in the MLP was cross entropy and the optimizer was Adam (KINGMA; BA, 2014). In the experiments described in Section 4.2, the MLP is trained with visual features vectors of 1024 dimensions extracted from RGB images using the GoogLeNet, where the output of the adaptive average pool 2D layer is the visual features vector of each image used for training.

The classification of the objects and place category of each node can be performed whenever necessary with the current state of the vector $\mathbf{c}_j$. The same could be done to recognize other semantic properties of nodes using, for example, new classifiers through transfer learning (as will be demonstrated in Section 5.1.2) or clustering methods.

4.1.4 Image Localization

To present the use of a generated map in other type of visual task, we use the consolidated visual features vector $\mathbf{c}_j$ and the average features distance vector $\boldsymbol{\delta}_j$ of each node j to localize RGB images on the map. The images have their features extracted using GoogLeNet, where the output of the adaptive average pool 2D layer is used as the visual features vector $\mathbf{x}$ of each image. The node with the highest activation $ac(D_\omega(\mathbf{x}, \mathbf{c}_j), \boldsymbol{\omega}_j)$ represents the estimated location of the image on the map:

$$loc(\mathbf{x}) = \arg\max_j [ac(D_\omega(\mathbf{x}, \mathbf{c}_j), \boldsymbol{\omega}_j)]. \tag{4.7}$$

Thus, $\boldsymbol{\omega}_j$ is the relevance vector of the node j, in which each component is computed by

an inverse logistic function based on the one proposed by Bassani and Araujo (2015):

$$\omega_{ji} = \begin{cases} \dfrac{1}{1 + exp\left(\frac{\delta_{ji} - \delta_{jimean}}{s(\delta_{jimax} - \delta_{jimin})}\right)}, & \text{if } \delta_{jimin} \neq \delta_{jimax} \\ 1, & \text{otherwise,} \end{cases} \tag{4.8}$$

where δ_{jimin}, δ_{jimax} and δ_{jimean} are respectively the minimum value, the maximum value and the average of the components of the distance vector $\boldsymbol{\delta}_j$. $s > 0$ controls the slope of the sigmoid function.

The activation $ac(D_\omega(\mathbf{x}, \mathbf{c}_j), \boldsymbol{\omega}_j)$ for each node j is calculated by a function of the weighted distance and the sum of the components of the relevance vector (introduced on (BASSANI; ARAUJO, 2012)):

$$ac(D_\omega(\mathbf{x}, \mathbf{c}_j), \boldsymbol{\omega}_j) = \frac{\|\boldsymbol{\omega}_j\|_1}{D_\omega(\mathbf{x}, \mathbf{c}_j) + \|\boldsymbol{\omega}_j\|_1 + \varepsilon}, \tag{4.9}$$

where ε is a small value to avoid division by zero, $\|.\|_1$ is the L^1-norm, and $D_\omega(\mathbf{x}, \mathbf{c}_j)$ is the weighted Euclidean distance between the consolidated visual features vector and the image visual features vector, with weights given by $\boldsymbol{\omega}_j$:

$$D_\omega(\mathbf{x}, \mathbf{c}_j) = \sqrt{\sum_{i=1}^{m} \omega_{ji}(x_i - c_{ji})^2}. \tag{4.10}$$

Such an adaptive distance is validated in (BASSANI; ARAUJO, 2015) and provides better results with high dimensional data. Furthermore, in this thesis $\boldsymbol{\delta}_j$ is only used for the task of topological image localization, but could be used to obtain semantic properties through unsupervised and semi-supervised subspace clustering methods (BRAGA; BASSANI, 2018), which is a future work.

4.2 EXPERIMENTS

This section details the experiments performed to evaluate the topological maps produced with the proposed method and their consolidated representations of deep visual features. First, Section 4.2.1 describes the dataset used in the experiments and Section 4.2.2 details how the hyperparameters of the method were adjusted for the experiments. Next, the topology of the generated maps is evaluated in Section 4.2.3. Then, the quality of the consolidated representations is evaluated in the tasks of object classification (Section 4.2.4), place classification (Sections 4.2.5 and 4.2.6), and topological image localization (Section 4.2.7).

4.2.1 Dataset

The experiments were performed using data selected from the COLD dataset[1](COsy Localization Database) (PRONOBIS; CAPUTO, 2009), a real-world indoor dataset that has three

[1] URL: https://www.cas.kth.se/COLD/

sub-datasets acquired in three different laboratories: COLD-Freiburg, COLD-Ljubljana, and COLD-Saarbrucken. Each sub-dataset contains data sequences acquired in different paths in the facilities. Not all data sequences in the dataset were used due to inaccuracies in the position data provided. We selected for use in experimentation 18 data sequences of 6 paths, 3 sequences of each path. Of the total, 6 data sequences of 2 paths are from COLD-Freiburg, where path 1 is a sub-part of path 2. In addition, 12 sequences of 4 paths are from COLD-Saarbrucken, in which paths 1 and 3 are, respectively, sub-parts of paths 2 and 4. The dataset has different types of data, but only RGB images from regular camera and their acquisition positions (given in meters) were used. The positions were computed by the dataset authors with a laser-based localization technique. In all selected data, there are 11 place categories, 12592 images in COLD-Freiburg and 17700 in COLD-Saarbrucken (examples of images from both sub-datasets in Figure 11). Table 1 lists the place categories and presents the number of images in each sub-dataset by place category.

Figure 11 – Examples of images from the COLD-Freiburg (a) and COLD-Saarbrucken (b) sub-datasets.

(a) (b)

Source: Pronobis and Caputo (2009).

4.2.2 Hyperparameter Adjustment

Almost all hyperparameters of the proposed method (and image localization approach) were tuned for the following experiments using a hyperparameter sampling technique, the Latin Hypercube Sampling (LHS) (HELTON; DAVIS; JOHNSON, 2005). The LHS samples the hyperparameters within pre-established ranges and the ranges of each hyperparameter are divided into subintervals of equal probability. Then, a single value is randomly chosen from each subinterval.

The spatial distance threshold λ was set manually and the value of 0.9 meters was used for all experiments. All other hyperparameters were sampled using LHS and we selected the

hyperparameter setting that provided the best result in each experimentation scenario, i.e., a different hyperparameter setting for each evaluation and dataset configuration. Table 2 presents the ranges used for each hyperparameter.

Table 1 – Total number of images in the selected data and by place category.

Place Category	COLD-Freiburg	COLD-Saarbrucken
Corridor	5732	7692
Printer area	1432	1413
Robotics lab	0	628
Stairs area	1023	0
Bathroom	1284	2362
Kitchen	483	643
Terminal room	0	1701
Conference room	0	1526
1-person office	511	1052
2-person office	1672	683
Large office	455	0
Total	12592	17700

Source: Created by the thesis author using data selected from Pronobis and Caputo (2009).

Table 2 – Hyperparameter ranges.

Hyperparameter	min	max
Features consolidation rate (α)	0.001	0.1
Minimum distance of features (τ)	1.0	100.0
Features persistence rate (γ)	0.001	0.999
Moving average rate (β)	0.001	0.999
Relevance smoothness (s)	0.001	0.1

Source: Thesis author.

4.2.3 Topology

In the proposed method, an adequate positioning of the topological nodes is vital for an adequate consolidation of the visual features. Thus, we evaluated the positioning of the nodes by checking if they were coherent with the spatial positions provided in the input data. For

this, the data sequences that were captured in the same area were presented randomly to the method and a single map was generated per area, where each map was generated with 6 data sequences of 2 paths. The selected data contains a single area (paths 1 and 2) in COLD-Freiburg, and in COLD-Saarbrucken it contains area 1 (paths 1 and 2) and area 2 (paths 3 and 4).

The process was repeated 10 times per area and in each execution the positions of the nodes in the generated map were first visually evaluated by plotting them over the positions of all input data (Figure 12 shows examples). Then, the mean and maximum Euclidean distances between all input position data and their nearest nodes were computed. The results obtained in all executions were averaged, resulting in: 0.304 ± 0.004 / 0.873 ± 0.010 (mean / maximum distance) for Freiburg single area; 0.314 ± 0.009 / 0.863 ± 0.020 (mean / maximum distance) for Saarbrucken area 1; and, 0.313 ± 0.022 / 0.842 ± 0.025 (mean / maximum distance) for Saarbrucken area 2.

The results allowed us to conclude that all the nodes were positioned coherently with the input data (with mean distance close to 0.3 meters) and the spatial distance threshold λ (with maximum distance always below $\lambda = 0.9$ meters). We also analyzed all connections between the nodes and verified that they all represented viable transitions in the environments.

4.2.4 Object Classification

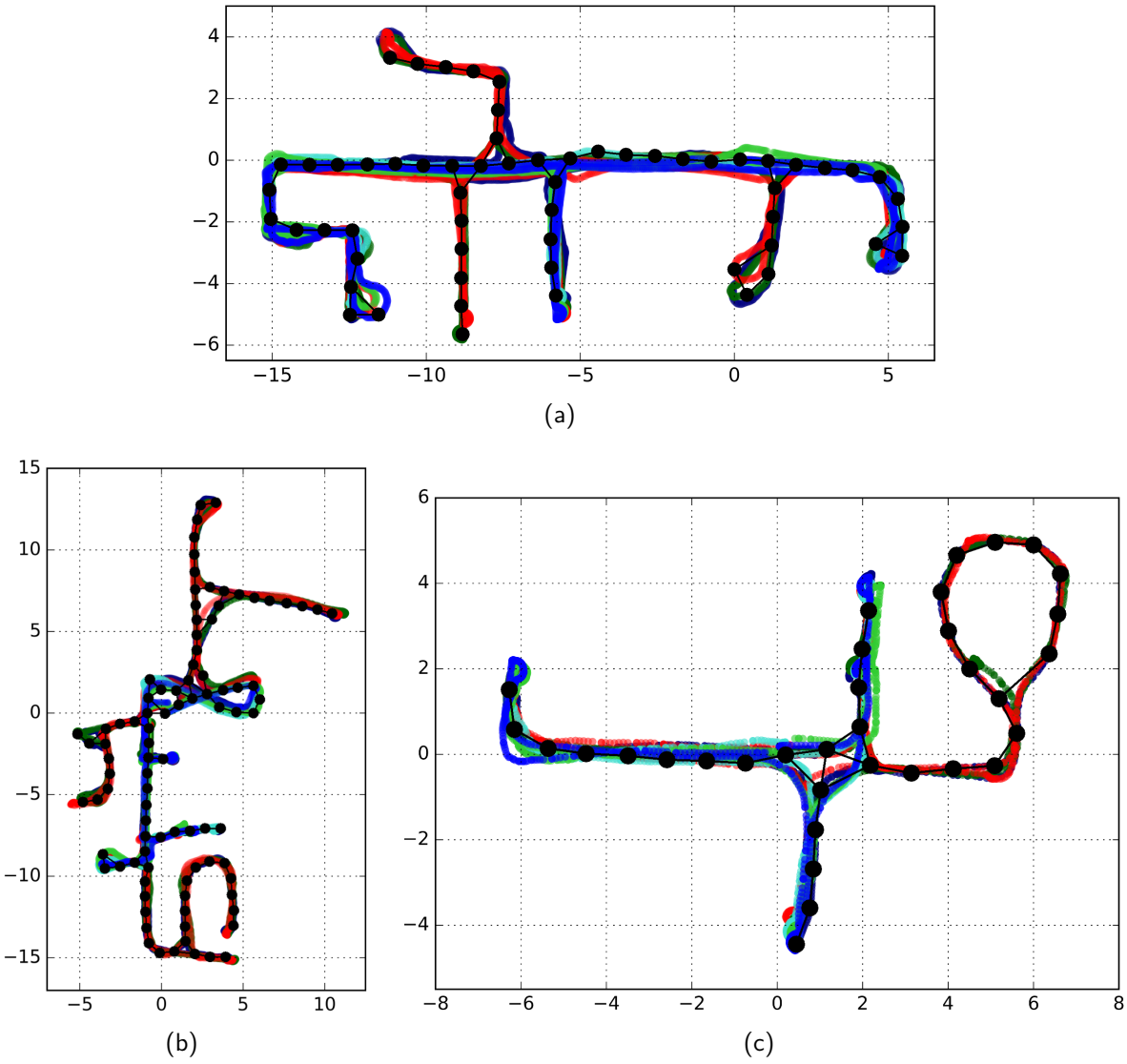
In this experiment, we assume that in the evaluated indoor environments only a small number of instances of classes are present and we use object classification as a sensible task to evaluate the quality of the consolidated visual features vectors with respect to preserving the specific visual characteristics of the original images they consolidated and, consequently, of the regions they cover. Since the visual features of multiple images are consolidated in the same node j, the TOP-1 label assigned by the GoogleNet to an image may not be the TOP-1 label when classifying the consolidated visual features vector $\mathbf{c}_j$ using the same classification layer (as described in Section 4.1.3). However, we expect it to be ranked close to the top. To evaluate this, we compute how frequently the TOP-1 classification label (among all the 1000 classes in ImageNet-1K[2]) of each image is present in the TOP-5 classification labels of $\mathbf{c}_j$ of each node j that consolidates the image. Images can be consolidated by more than one node as long as they have been captured at an intersection position between nodes.

This evaluation is done per sub-dataset, with all selected data sequences from all paths and one map generated per data sequence. Table 3 presents the accuracy results obtained, which suggest that the consolidated visual features fairly preserve the classifications observed in the original images.

As the TOP-5 accuracy is not sensitive enough to detect small variations, to evaluate the effect of the visual persistence (VP) capability of the consolidation process, we classify

[2] Most classes in ImageNet-1K are objects, but they also include some breeds of animals.

Figure 12 – Nodes of topological maps generated with the selected data sequences from COLD-Freiburg single area (a), from COLD-Saarbrucken area 1 (b), and from COLD-Saarbrucken area 2 (c). The nodes were plotted over the positions of all the input data and the selected data sequences from each area were provided to the proposed method in random order. Each colored line (there are six colors) represents the positions provided in a different data sequence. The nodes of the map and their connections are in black.



Source: Thesis author.

the vector $\mathbf{c}_j$ of each node j with the linear classification layer of the GoogLeNet, provide the outputs of 13 classes of objects from ImageNet-1K (selected based on what is found in the COLD images[3]) for a softmax layer and obtain the classification vector $\mathbf{o}_j$. Then, we obtain the average classification of all images (classified by GoogLeNet, with the outputs of the same 13 classes provided for a softmax layer) captured around each node j in the vector $\mathbf{m}_j$. Finally, we compare $\mathbf{m}_j$ to $\mathbf{o}_j$ for all nodes with an error measure based on the l1-norm: $err_{mean} = \frac{1}{b \times k} \sum_{j=1}^{k} ||o_j - m_j||_1$, where $b = 13$ is the number of classes of objects we selected

[3] Object classes selected for the evaluation: washbasin, soap dispenser, toilet seat, photocopier, monitor, desktop computer, desk, dining table, barber chair, microwave oven, stove, dishwasher and toaster.

and k is the sum of the number of nodes in each generated map. This evaluation is also done per sub-dataset, with one map generated per data sequence.

Table 3 presents the results obtained, which give evidence of the positive contribution of the visual features persistence in the consolidation process. The results also provide further evidence that the consolidated visual features are representative of their regions in order to approximate the classification results to those of the average classification of all images captured in those regions.

Table 3 – Results of the objects classification evaluation.

	COLD-Freiburg	COLD-Saarbrucken
$TCMA^{*}(Accuracy)$	0.8185	0.7616
$TCMA\ (err_{mean})$	0.0123(0.0068)	0.0163(0.0079)
$TCMA_{-VP}\ (err_{mean})$	0.0146(0.0083)	0.0196(0.0109)

Standard deviations are shown in parentheses.
* $TCMA$ is the proposed method with the complete TCMA process and $TCMA_{-VP}$ is the proposed method with the TCMA process without Visual Persistence.

Source: Thesis author.

4.2.5 Place Classification

In this experiment, the consolidated visual features vector $\mathbf{c}_j$ of each node j is fed into the MLP described in Section 4.1.3 to classify the nodes in one of the 11 place categories present in the dataset. This experiment is performed using an adaptation of the k-fold cross-validation method, in which each fold is a sequence of data selected from the datasets, k is the number of sequences chosen, and the order of the sequences is randomly selected. Thus, all images in the $k-1$ sequences of data are used to train the MLP in batch mode, with all images per epoch. The number of epochs is experimentally determined and only one value is defined for each cross-validation run. The remaining data sequence is used in the generation of the map and the resulting consolidated visual features vector $\mathbf{c}_j$ of each node j is classified by the MLP. Therefore, this experiment is done per area, i.e., we use per execution the selected data sequences that were captured in the same area (for instance, the data sequences from paths 1 and 2 of COLD-Saarbrucken area 1).

Table 4 presents the excellent results obtained (classification accuracy), which suggest that even a simple MLP can be used to accurately recognize the place categories of the nodes from the visual features consolidated in them and that the consolidated representations preserve the specific visual characteristics of the regions they cover. Figure 13 illustrates an example of the place classification results obtained with the map generated from a data sequence of

COLD-Freiburg, in which we can see that only two nodes were misclassified. Both nodes are located close to the borders between different place categories in the ground truth, from where it is possible to see the other regions (Figure 13 exemplifies images), what may have led to the errors.

For comparison, we used the same adaptation of the cross-validation method with the difference that, at each run, the visual features vectors extracted from all images of the remaining data sequence are directly classified by the MLP. Again, the number of training epochs was experimentally determined and only one value was defined for each cross validation run. The results obtained (*IMAGES* columns in Table 4) were all smaller than the results obtained with the proposed method, which suggests that the proposed features consolidation process produces rich visual representations of regions and contributes positively to the classification of place categories.

Table 4 – Results of the place classification experiment. Standard deviations are shown in parentheses.

Accuracy	$TCMA$[1]			*IMAGES*		
	Fr1 and Fr2[2]	Sa1 and Sa2	Sa3 and Sa4	Fr1 and Fr2	Sa1 and Sa2	Sa3 and Sa4
Corridor	0.987(0.018)	0.984(0.016)	1.000(0.000)	0.975(0.011)	0.958(0.021)	0.964(0.007)
Printer area	1.000(0.000)	1.000(0.000)	1.000(0.000)	0.914(0.057)	0.908(0.072)	0.972(0.025)
Robotics lab	—[3]	0.944(0.078)	—	—	0.665(0.013)	—
Stairs area	0.944(0.124)	—	—	0.911(0.032)	—	—
Bathroom	1.000(0.000)	1.000(0.000)	1.000(0.000)	0.972(0.025)	0.965(0.012)	0.978(0.009)
Kitchen	1.000(0.000)	—	0.970(0.043)	0.730(0.147)	—	0.914(0.056)
Terminal room	—	0.968(0.022)	—	—	0.917(0.071)	—
Conference room	—	1.000(0.000)	—	—	0.921(0.065)	—
1-person office	1.000(0.000)	0.933(0.094)	1.000(0.000)	0.845(0.091)	0.818(0.031)	0.977(0.021)
2-person office	0.907(0.099)	0.917(0.186)	—	0.891(0.104)	0.861(0.054)	—
Large office	1.000(0.000)	—	—	0.699(0.072)	—	—
Overall	0.979(0.021)	0.982(0.013)	0.995(0.011)	0.931(0.025)	0.928(0.032)	0.967(0.015)

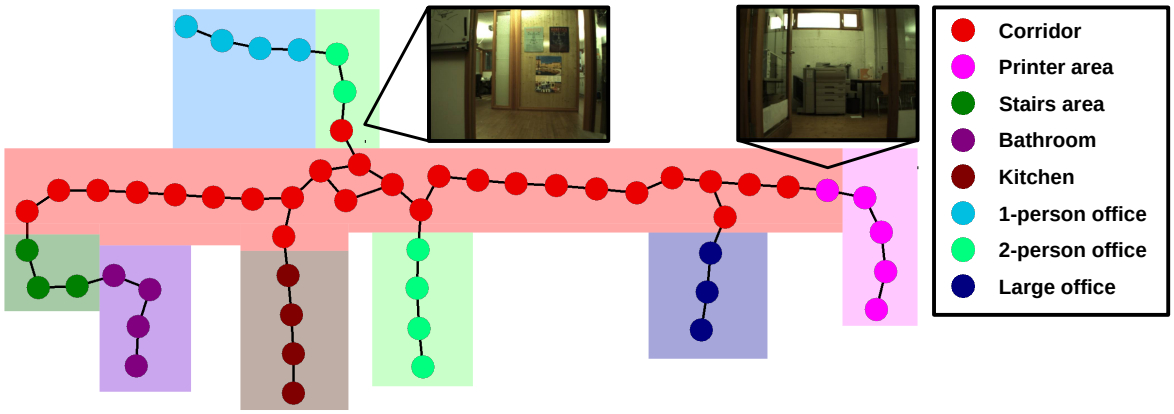
[1] $TCMA$ stands for the proposed method and *IMAGES* denotes the results obtained with the direct classification of the visual features vectors extracted from the images.
[2] Fr(n) is Freiburg (n = path 1 or 2) and Sa(n) is Saarbrucken (n = path 1, 2, 3 or 4).
[3] — means the place category is not present in the paths used.

Source: Thesis author.

In addition, we performed a complementary comparison with the results of place classification obtained by Rubio et al. (2016) and Mancini et al. (2017). Both works use the COLD dataset in a stratified 5-fold cross-validation process using only the path 2 images of each sub-dataset, separately. They consider each room as a place category and as there are two different 2-person office rooms in the path 2 of Freiburg, we use 9 categories for the Freiburg path 2 in this comparison only. The number of place categories used for the Saarbrucken path

2 remains 8. In (RUBIO et al., 2016), the authors extract HOG (Histogram of Oriented Gradients) features from RGB images and perform the classification using different classifiers. The best results from Rubio et al. (2016) were obtained using a Bayesian network as classifier and we used these results for comparison. Mancini et al. (2017) introduce an end-to-end trainable deep approach to place classification that integrates a NBNN (Naive Bayes Nearest Neighbor) model into a CNN.

Figure 13 – Example of place classification results obtained with the map generated from a path 2 data sequence of COLD-Freiburg. The color of each node is the class assigned by the MLP and the colored blocks represent the place categories of the nodes covered by the blocks in the ground truth. In the black boxes are examples of images captured in the regions of each misclassified node, these images demonstrate how visible is specific visual data of other place categories (e.g., a printer at the entrance to the printer area) from the misclassified nodes.



Source: Thesis author.

In order to approximate the experimentation scenarios for comparison, we use the same cross validation adaptation used earlier with the proposed method and consider only the selected data from path 2 of each sub-dataset. Table 5 presents the results obtained (denoted here as room classification), which are higher than the best obtained by Rubio et al. (2016) and similar to the results of Mancini et al. (2017). This suggests that the visual features consolidated with the proposed method can also be used to recognize the specific room of the topological nodes with similar accuracy to the current literature.

Table 5 – Room classification comparison with Rubio et al. (2016) and Mancini et al. (2017).

Accuracy	Fr2	Sa2
Rubio et al. (2016)	0.823[1]	0.844
Mancini et al. (2017)	0.952	0.973
$TCMA$	0.957(0.018)	0.973(0.011)

Standard deviations (STD) are shown in parentheses.
[1] Rubio et al. (2016) and Mancini et al. (2017) did not provide variance or STD values.

Source: Created by the thesis author including data from Rubio et al. (2016) and Mancini et al. (2017).

4.2.6 Place Classification Over Time

In this experiment, from each area in the dataset, we first selected data sequences of the longest path that contain all the place categories found in that respective area. Then, in an evaluation procedure repeated 10 times, we use one of these data sequences to train the MLP and the remaining 5 data sequences from the same area are used to generate the map. In the generation process, when all data in a data sequence is finished, the current state of the consolidated features vector $\mathbf{c}_j$ of each node j is classified by the MLP. Thus, the evaluation takes place in 5 instants of time in the mapping process. This evaluation procedure is performed with each data sequence selected from the longest paths and the order of the remaining data sequences used in the generation of the maps is randomized at each run.

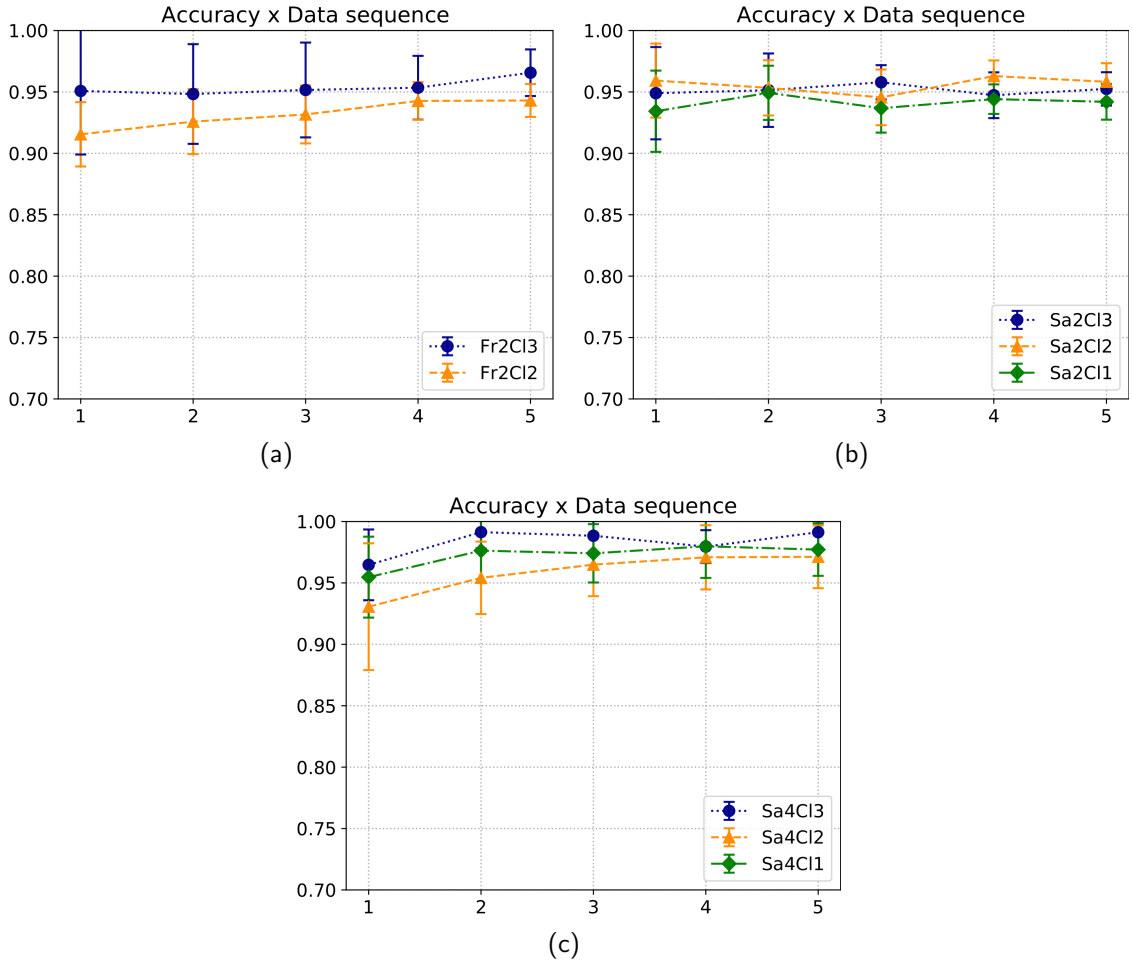
The mean and standard deviation of the accuracy results obtained at each instant of time are illustrated in the graphs in Figure 14. They show that for all three areas the results are stable or increasing as more data is consolidated in the mapping process. This suggests that the consolidated representations do not degrade with more data or might even improve. Figure 14 also shows that the standard deviations tend to get smaller as more data is consolidated into the maps, suggesting that the consolidated representations are less biased in relation to the initial features received and more representative of the regions covered by each topological node.

4.2.7 Image Localization Experiment

In the last experiment of this chapter, we use the visual data consolidated in each node to localize images on the maps using the approach described in Section 4.1.4 and study the effect of the visual habituation (VH) capability of the consolidation process. This experiment is performed using the same adaptation of the k-fold cross-validation method used earlier with the proposed method in Section 4.2.5, but here $k-1$ sequences of data are used in the generation of the map and all images in the remaining sequence of data are localized on the map. Thus, in this experiment we use per execution only the selected data sequences that were captured in the same area and a single map is generated per area.

We evaluated this in three scenarios in which 10% of the images in each sequence were replicated 0 times in the first scenario, 20 in the second, and 40 times in the third. The images to be replicated were randomly selected at the beginning of the map construction in each execution from the data sequences used to build the map. With this, we simulate moments when the robot stops moving and captures very similar images for a while. Table 6 shows the TOP-1 and TOP-5 accuracy values measured with the proposed method with the complete TCMA process ($TCMA$) and with the TCMA process without visual habituation capability ($TCMA_{-VH}$). TOP-1 evaluates whether for each image the node on the map with the highest activation is the node (or one of the nodes) in the ground truth. TOP-5 evaluates whether,

Figure 14 – Results of the place classification over 5 instants of time: (a) shows the average results obtained with the data from Freiburg area; (b) and (c) show the average results obtained with the data from Saarbrucken area 1 and 2, respectively. Each legend item is the data sequence used to train the MLP. Fr2Cl1 was not used in (a) because it does not contain all the place categories.



Source: Thesis author.

for each image, one of the five nodes on the map with the highest activation is the node (or one of the nodes) in the ground truth. The ground truth for an image can be more than one node, if it was captured at an intersection position between nodes.

The results with the complete consolidation process ($TCMA$) remained stable as the number of image replications increased, which shows the positive influence of the visual habituation capability in the consolidation of visual features. While the results in $TCMA_{-VH}$ got worse as the number of image replications increased, which suggests that without the visual habituation capability it would increasingly degrade its consolidated features the longer the robot stays put, due to the sequential presentation of similar features. In addition, the results in $TCMA$ suggest that the consolidated representations in each node can be used to locate images with good accuracy.

Table 6 – Results of the image localization experiment with random image replication. Standard deviations are shown in parentheses.

IR[1]	Data Sequences	$TCMA$[2]		$TCMA_{-VH}$	
		TOP-1 Acc	TOP-5 Acc	TOP-1 Acc	TOP-5 Acc
0	Fr1(Cl1,Nt1,Nt3), Fr2(Cl1,Cl2,Cl3)[3]	0.744(0.067)	0.907(0.033)	0.741(0.036)	0.912(0.014)
	Sa1(Cl1,Cl2,Cl3), Sa2(Cl1,Cl2,Cl3)	0.697(0.059)	0.911(0.028)	0.700(0.072)	0.907(0.034)
	Sa3(Cl1,Cl2,Cl3), Sa4(Cl1,Cl2,Cl3)	0.834(0.021)	0.976(0.016)	0.837(0.015)	0.973(0.017)
20	Fr1(Cl1,Nt1,Nt3), Fr2(Cl1,Cl2,Cl3)	0.746(0.066)	0.904(0.019)	0.686(0.076)	0.883(0.054)
	Sa1(Cl1,Cl2,Cl3), Sa2(Cl1,Cl2,Cl3)	0.693(0.065)	0.918(0.037)	0.672(0.076)	0.893(0.043)
	Sa3(Cl1,Cl2,Cl3), Sa4(Cl1,Cl2,Cl3)	0.846(0.024)	0.976(0.018)	0.790(0.040)	0.968(0.021)
40	Fr1(Cl1,Nt1,Nt3), Fr2(Cl1,Cl2,Cl3)	0.759(0.060)	0.902(0.020)	0.638(0.068)	0.880(0.030)
	Sa1(Cl1,Cl2,Cl3), Sa2(Cl1,Cl2,Cl3)	0.693(0.052)	0.913(0.030)	0.650(0.079)	0.882(0.042)
	Sa3(Cl1,Cl2,Cl3), Sa4(Cl1,Cl2,Cl3)	0.833(0.043)	0.976(0.016)	0.753(0.061)	0.951(0.017)

[1] IR is nº of image replications and Acc stands for Accuracy.
[2] $TCMA$ is the proposed method with the complete TCMA process and $TCMA_{-VH}$ is the proposed method with the TCMA process without Visual Habituation.
[3] Fr(n) is Freiburg (n = path 1 or 2), Sa(n) is Saarbrucken (n = path 1, 2, 3 or 4), Cl(n) is cloudy sequence (n = sequence 1, 2 or 3) and Nt(n) is night sequence (n = sequence 1 or 3).

Source: Thesis author.

4.3 SUMMARY

This chapter presented a topological semantic mapping method that consolidates deep visual features of regions using a process we denoted as Topological Consolidation of Features by Moving Averages (TCMA). The process uses exponential moving averages and is empowered with visual persistence and visual habituation capabilities. The experiments, performed using a real-world indoor dataset, suggested that the consolidated representations are rich visual representations of the topological regions they cover and fairly represent the visual features of the original images they consolidated. Furthermore, the experiments suggested that the consolidated representations do not degrade over time, can be used for extracting different semantic properties (such as place category, specific room, and objects) and to indicate the topological location of images.

Despite of the promising results, such use of reusable consolidated representations of visual features for the acquisition of semantics for spatial maps represents an innovation for the area. Therefore, evaluating them in more practical scenarios is essential to demonstrate their applicability in real-world tasks and their capacity for flexible use in adapting the semantics of the maps to new requirements of new tasks.

With this purpose, the next chapter (Chapter 5) adapts the proposed method for real-time operation and uses classifiers trained on large-scale datasets from the literature to recognize different semantic properties for the same maps, generated from 5 different home environ-

ments and two scenarios of hyperparameter configuration. The first scenario uses the same hyperparameter setting for all environments and the second uses the best setting found for each environment. The chapter also presents a new visual features consolidation process based on the use of multiple vectors of moving averages per node.

This thesis uses the term *real-time* as a reference for the *online* execution of the method simultaneously with data capture in a real or simulated embedded scenario (regardless of the frame rate). The study of the maximum frame rate at which the method can be executed on current hardware equipment is not the focus of this thesis. However, for information, the experiments detailed in the next chapter were performed with the input data at a frame rate of approximately 5Hz on a laptop from the year 2013 with an Intel i7-4500U processor, 8GB of DDR3 RAM and NVIDIA GeForce GT 750M graphics card with 2GB of memory.

5 REAL-TIME MAPPING WITH FLEXIBLE SEMANTIC RECOGNITION

This chapter introduces a version for operation in real-time of the topological semantic mapping method introduced in the previous chapter. First, the operation of the method is detailed in Section 5.1. Then, a new features consolidation process based on the combination of multiple vectors of exponential moving averages is introduced in Section 5.2. Next, Section 5.3 introduces a modification of the method that uses a more classical semantic mapping approach based in the topological counting (or accumulation) of semantic properties recognized from individual images. Finally, Section 5.4 describes the experiments performed to evaluate the method and all its variations in more practical scenarios. The experiments are performed with data from 5 different home environments and in two scenarios of hyperparameter configuration.

5.1 THE METHOD FOR REAL-TIME OPERATION

The topological semantic mapping method proposed in the previous chapter (Chapter 4) was adapted for real-time operation in an integrated solution built on ROS (Robot Operating System) (QUIGLEY et al., 2009). The solution uses the ORBSLAM2 (MUR-ARTAL; TARDOS, 2017) for 3D metric SLAM (Simultaneous Localization and Mapping) and for deep features extraction a CNN model designed for mobile and resource-constrained environments, the MobileNetV2 (SANDLER et al., 2018), pre-trained on ImageNet-1K and available in the Pytorch library.

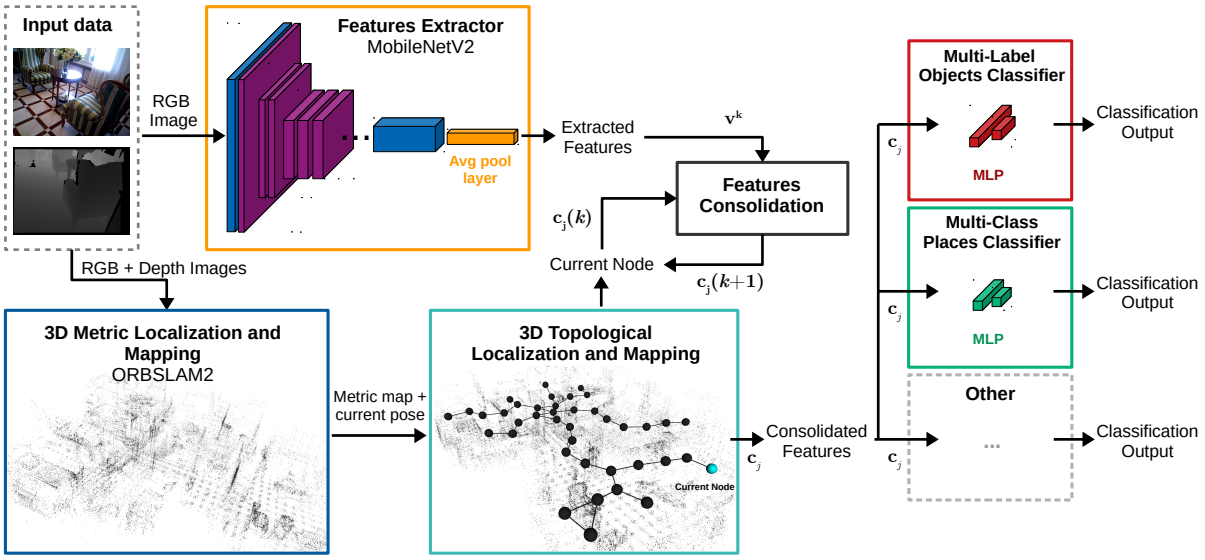
Two shallow MLP classifiers recognize semantic properties from the visual features consolidated in each topological node. Both MLPs are trained on large-scale datasets from the literature using the pre-trained MobileNetV2 for transfer learning. The first is a multiclass classifier that recognizes the nodes in 6 different place categories. The second is a multi-label classifier that indicates the presence of 10 different objects classes in the regions covered by each topological node. An overview of the real-time version of method is illustrated in Figure 15. The following sections detail how the method operates, highlighting similarities and differences with the previous version.

5.1.1 Topological Consolidation of Visual Features

The method receives as input RGB and Depth images, asynchronously. At each time step k (following the RGB images frequency), the 3D Metric SLAM module (ORBSLAM2) estimates the current spatial position $\mathbf{s}^k = \{s_i^k, i = 1...n\}$, where $n = 3$. MobileNetV2 receives as input the RGB image $\mathbf{z}^k$, the output of the 2D adaptive average pooling layer is flattened in the vector $\mathbf{v}^k$ of $m = 1280$ dimensions and used as a visual features vector.

The rest of the method follows almost the same process described earlier in Sections

Figure 15 – Overview of the method: ORBSLAM2 estimates the current spatial position $\mathbf{s}^k$ on the metric map for the input images. MobileNetV2 extracts the visual features vector $\mathbf{v}^k$ of the input RGB images, and the nodes in the topological map that correspond to the obtained metric spatial positions consolidate $\mathbf{v}^k$. The consolidated visual features vector $\mathbf{c}_j$ of each node is classified by two shallow MLPs trained on large-scale datasets of the literature using the pre-trained MobileNetV2 for transfer learning. The first indicates the place category of each node out of 6 available and the second, a multi-label classifier, indicates the presence of 10 different object classes in the region covered by each node. Other classifiers could be added to recognize different semantic properties, they just need to be trained using the same pre-trained CNN, the MobileNetV2 pre-trained on ImageNet-1K.



Source: Thesis author.

4.1.1 and 4.1.2. Each node j in the topological map has a spatial position vector $\mathbf{p}_j = \{p_{ji}, i = 1...n\}$ and a consolidated visual features vector $\mathbf{c}_j = \{c_{ji}, i = 1...m\}$. As $\mathbf{v}^k$ and $\mathbf{s}^k$ are obtained, the current topological location is determined by the node with the smallest Euclidean distance between the given spatial position $\mathbf{s}^k$ and the spatial position of the node $\mathbf{p}_j$, if the distance is equal to or smaller than the spatial distance threshold λ. If not, or if the map is empty, a new node η is inserted into the topological map.

Node η is initialized with $\mathbf{p}_\eta = \mathbf{s}^k$ and the current state of the consolidated visual features vector $\mathbf{c}_l$ of the last visited node l, if any, is persisted in $\mathbf{c}_\eta$ through an average between the input visual features $\mathbf{v}^k$ and $\mathbf{c}_l$ determined by γ, the persistence rate (as per Equation 4.3).

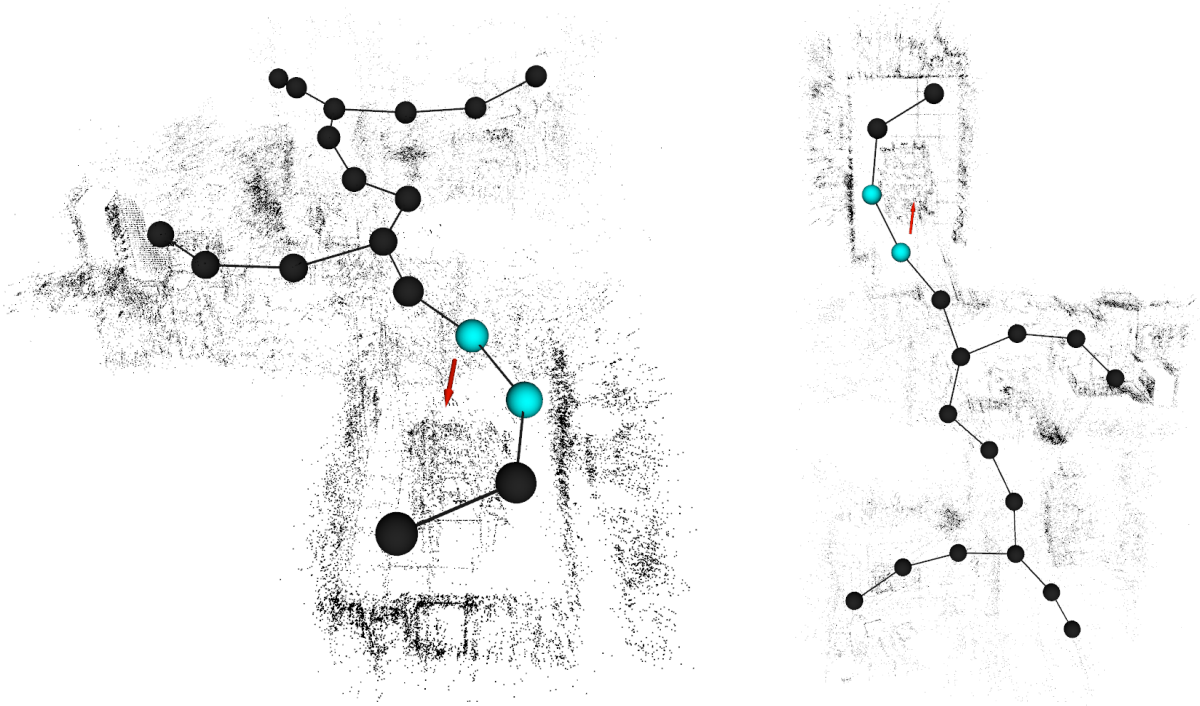
If the current topological location was determined by the node with the smallest Euclidean distance to $\mathbf{s}^k$, the node consolidates $\mathbf{v}^k$ using a modification of the exponential moving average detailed in Equation 4.4. In the modification, the squared Euclidean distance is replaced by the Manhattan distance (also known as L1-norm) as the distance metric used to control the visual habituation of the node, where $\mathbf{v}^k$ is consolidated by the node only if the distance $D(\mathbf{v}^k, \mathbf{u}_j)$ between the input visual features vector $\mathbf{v}^k$ and the last visual features vector consolidated in the node, $\mathbf{u}_j$, is equal to or greater than the minimum distance τ. The Manhattan distance was chosen as it works better on high dimensional data than the squared Euclidean distance.

Therefore, $D(\mathbf{v}^k, \mathbf{u}_j)$ is now defined as follows:

$$D(\mathbf{v}^k, \mathbf{u}_j) = \sum_{i=1}^{m} |v_i^k - u_{ji}|. \tag{5.1}$$

Then, other close nodes also consolidate $\mathbf{v}^k$ following the same process if their Euclidean distances to $\mathbf{s}^k$ are also equal to or smaller than λ. Figure 16 illustrates the process when two nodes consolidate the same visual features vector $\mathbf{v}^k$ captured in a spatial position $\mathbf{s}^k$ that attends λ for both nodes. Finally, as in the previously introduced version of the method, a connection between nodes is created whenever a transition occurs between them.

Figure 16 – Illustrations of two topological nodes (in cyan color) consolidating the same visual features vector captured in a metric spatial position (red arrow) that attends the spatial distance threshold λ for both nodes. (a) shows a close-up view and (b) a top view. Metric and topological maps were generated for the path home_at of the SUN3D dataset (XIAO; OWENS; TORRALBA, 2013), described in Section 5.4.2.3.



(a) (b)

Source: Thesis author.

5.1.2 Semantic Properties Classification

In the method, the consolidated visual features vector $\mathbf{c}_j$ of each topological node is classified by two MLPs (implemented using the PyTorch library) that indicate the place category of the node out of 6 available and the presence of 10 different objects in the region covered by the node. The pre-trained MobileNetV2 was used in the training process of both classifiers. For this, all the weights of the MobileNetV2 were frozen except the ones of the linear classification layer.

For the places classifier, the linear classification layer (i.e., the layer right after the 2D adaptive average pooling layer) was replaced by an MLP (with random weights) that contains one hidden layer with 48 units, followed by ReLU activation functions and 6 units in the output layer. Then, the model was trained and validated with the data selected from the Places365-Standard dataset (ZHOU et al., 2018) (details in Section 5.4.2.1). The loss function used for training was cross entropy and the optimizer was Adam (KINGMA; BA, 2014). Furthermore, the dropout and batch normalization layers were kept active during training. Finally, the trained MLP was detached from the CNN and used as places classifier.

For the multi-label objects classifier, the MLP that replaces the linear classification layer contains one hidden layer with 160 units, ReLU activation functions and 10 units in the output layer. The model was trained and validated with the data selected and prepared from the COCO dataset (LIN et al., 2014) (details in Section 5.4.2.2). The optimizer was the same (Adam) and the loss function used was binary cross entropy. Again, the dropout and batch normalization layers were kept active during training. Lastly, the trained MLP was detached from the CNN and used as multi-label objects classifier. The learning rate and number of epochs of training were determined experimentally for both classifiers.

The same process could be done to train and include other classifiers in the future. These classifiers could, from the same vectors of consolidated visual features (i.e., the vector $\mathbf{c}_j$ of each topological node), for example, add new semantic properties to the map, add new semantic classes (e.g., a new place category), or even replace old classifiers for better performance.

5.2 VISUAL FEATURES CONSOLIDATION BY THE COMBINATION OF MULTIPLE MOVING AVERAGES

This section introduces a new process of visual features consolidation that calculates the consolidated visual features vector $\mathbf{c}_j$ of a node j by the combination, through the arithmetic mean, of multiple vectors of exponential moving averages consolidated in the node. We denote this process as Topological Consolidation of Features by the Combination of Multiple Moving Averages (TCCMA) and the main purpose of the introduction of this process is to serve as a comparison to the TCMA process (introduced in Section 4.1) in the experiments, where the method is evaluated using the two processes separately.

In the process, in addition to the consolidated visual features vector $\mathbf{c}_j = \{c_{ji}, i = 1...m\}$ it is added to each node j a vector of consolidation items $\mathbf{I}_j = \{\mathbf{I}_{ji}, i = 1...q\}$, where each $\mathbf{I}_{ji} = \{I_{jil}, l = 1...m\}$ is a exponential moving averages vector and q is a number that changes for each node.

First, when a new node is created in the topological map, the current state of the consolidated visual features vector $\mathbf{c}_l$ of the last visited node l, if any, is persisted in the first consolidation item $\mathbf{I}_{j\eta}$ created for $\mathbf{I}_j$ through an average between the input visual features

vector $\mathbf{v}^k$ and $\mathbf{c}_l$, as follows:

$$\mathbf{I}_{j\eta} = \begin{cases} \gamma \mathbf{c}_l + (1-\gamma)\mathbf{v}^k, & \text{if there is a } \mathbf{c}_l \\ \mathbf{v}^k, & \text{otherwise,} \end{cases} \tag{5.2}$$

where γ is the persistence rate.

Then, as in the TCMA, the input visual features vector $\mathbf{v}^k$ is consolidated by a node j only if the distance $D(\mathbf{v}^k, \mathbf{u}_j)$ (Equation 5.1) between $\mathbf{v}^k$ and the last visual features vector consolidated in the node, $\mathbf{u}_j$, is equal to or greater than τ, which represents the visual habituation control of the node. If $D(\mathbf{v}^k, \mathbf{u}_j) >= \tau$, the consolidation items in $\mathbf{I}_j$ compete to consolidate $\mathbf{v}^k$ and the winner is the item with the smallest distance to $\mathbf{v}^k$:

$$h(\mathbf{v}^k) = \arg\min_i [D(\mathbf{v}^k, \mathbf{I}_{ji})], \tag{5.3}$$

where $D(\mathbf{v}^k, \mathbf{I}_{ji})$ is the Manhattan distance described as follows:

$$D(\mathbf{v}^k, \mathbf{I}_{ji}) = \sum_{l=1}^{m} |v_l^k - I_{jil}|. \tag{5.4}$$

If the distance of the winner to $\mathbf{v}^k$ is greater than the distance threshold of consolidation items ζ, then a new consolidation item η is added in $\mathbf{I}_j$ with $\mathbf{I}_{j\eta} = \mathbf{v}^k$. If not, the winning consolidation item is updated in direction to $\mathbf{v}^k$ as follows:

$$\mathbf{I}_{ji}(k+1) = \mathbf{I}_{ji}(k) + \alpha(\mathbf{v}^k - \mathbf{I}_{ji}(k)). \tag{5.5}$$

After the update of the consolidation items vector $\mathbf{I}_j$, the consolidated visual features vector $\mathbf{c}_j$ is updated by the arithmetic mean of all consolidation items in $\mathbf{I}_j$, as follows:

$$\mathbf{c}_j(k+1) = \tfrac{1}{q} \sum_{i=1}^{q} \mathbf{I}_{ji}(k+1). \tag{5.6}$$

5.3 TOPOLOGICAL SEMANTIC MAPPING BY INDIVIDUAL IMAGE CLASSIFICATION COUNTING

This section presents a modification of the proposed method that uses an approach based on topological counting (or accumulation) of semantic properties recognized from individual images. The modification follows a more classical semantic mapping approach and is only used as a simple baseline method for comparison in the experiments (Sections 5.4.4 and 5.4.5), where it's use is always referred to as the TC (Topological Counting) method.

In the modification, the consolidation process is removed and each topological node j contains two vectors: the place category counts vector $\mathbf{q}_j$ of 6 dimensions (one for each place category), and the objects presence counts vector $\mathbf{g}_j$ of 10 dimensions (one for each object label).

During the mapping process, each visual features vector $\mathbf{v}^k$ extracted by MobileNetV2 is classified by the places classifier and the multi-label objects classifier (introduced in Section 5.1.2). Each node j with Euclidean distance between the spatial position vector $\mathbf{p}_j$ and the current spatial position $\mathbf{s}^k$ equal to or smaller than λ, sums one to the value in the dimension of the place category counts vector $\mathbf{q}_j$ corresponding to the place category indicated by the classifier (given in numeric value):

$$q_{ji}(k+1) = q_{ji}(k) + 1, \tag{5.7}$$

where i is the place category indicated by the classifier.

Also, in the same way, each node j with Euclidean distance to $\mathbf{s}^k$ equal to or smaller than λ, sums each dimension of the objects presence counts vector $\mathbf{g}_j$ to each dimension of the 10-dimensional binary output vector $\mathbf{o}$ given by the multi-label objects classifier, as follows:

$$\mathbf{g}_j(k+1) = \mathbf{g}_j(k) + \mathbf{o}. \tag{5.8}$$

At the end of the mapping process, for each node j, the dimension of the place category counts vector $\mathbf{q}_j$ with the highest value indicates the place category of the node and each dimension in the objects presence counts vector $\mathbf{g}_j$ that has value higher than the objects count threshold determines the presence of the respective object label in the node. Therefore, the value set for the objects count threshold determines which object labels are present in each node.

Note that this modification of the proposed method does not provide the flexibility of the reusable consolidated representations and is included here only as a reference of quality, as it directly classifies the extracted feature vectors (the ideal scenario for the classifiers, which are trained from feature vectors of individual images).

5.4 EXPERIMENTS

This section details the experiments performed to evaluate the application of the proposed method in more practical scenarios. The method is evaluated in home environments with the consolidated visual features produced with both the TCMA and TCCMA processes, separately. First, the metrics used to evaluate the results in the experiments are described in Section 5.4.1. Then, the experimental setup is detailed in Section 5.4.2. Next, the topology of the produced maps is evaluated in Section 5.4.3. Finally, the consolidated representations are evaluated in two different hyperparameter scenarios for the simultaneous recognition of place categories (Section 5.4.4) and multiple object labels (Section 5.4.5), as semantic properties of the topological nodes.

5.4.1 Metrics

Two very common metrics in the literature are used to evaluate the results in the experiments. As in the previous chapter, accuracy ($\frac{No.\ of\ correct\ predictions}{Total\ no.\ of\ predictions}$) is used to evaluate the place classification results obtained in the experiments. Differently, the metric used to evaluate the multi-label object classification results is the F1-score, which is given by the harmonic mean of precision and recall:

$$F_1 = \frac{2 \cdot precision \cdot recall}{precision + recall}, \tag{5.9}$$

where $precision = \frac{No.\ of\ correct\ positive\ predictions}{No.\ of\ positive\ predictions}$ and $recall = \frac{No.\ of\ correct\ positive\ predictions}{No.\ of\ positive\ instances\ in\ the\ dataset}$.

However, as the objects classification results obtained are multi-label, a metric that considers all the output labels is more adequate to obtain an average perspective of the quality of the results. For this, the Macro Average F1-score is used as the main metric to evaluate the multi-label objects classification results obtained, which is calculated as follows:

$$\text{Macro Avg.} = \frac{1}{n} \sum_{i=1}^{n} F_{1i}, \tag{5.10}$$

where n is the number of labels considered in the average, which is 10 in the standard Macro Average F1-score in the experiments. However, a different version that considers only the labels present in the evaluated environments is also used in the experiments and is denoted as Macro Average F1-score B.

The data selected for the experiments are unbalanced, therefore, the Macro Average F1-score is a suitable metric to use. However, for comparison, the Micro Average F1-score is also used to evaluate the multi-label objects classification results. The metric can be calculated as the harmonic mean of Micro Average Precision and Micro Average Recall, which are calculated as follows:

$$\text{Micro Avg. Precision} = \frac{\sum_{i=1}^{n} No.\ of\ correct\ positive\ predictions\ of\ label\ i}{\sum_{i=1}^{n} No.\ of\ positive\ predictions\ of\ label\ i}, \tag{5.11}$$

$$\text{Micro Avg. Recall} = \frac{\sum_{i=1}^{n} No.\ of\ correct\ positive\ predictions\ of\ label\ i}{\sum_{i=1}^{n} No.\ of\ positive\ instances\ of\ label\ i\ in\ the\ dataset}, \tag{5.12}$$

where n is the number of labels considered in the average, which is 10 in the experiments.

5.4.2 Experimental Setup

The experiments demanded extensive preparation due to the more practical nature of the evaluations. Therefore, this section was created to detail all the information related to the experimental setup, such as the datasets used, the preparations made on the selected data, the annotation processes performed, and how the hyperparameters were adjusted.

5.4.2.1 *Places365 dataset*

The Places365-Standard was the dataset used to train the places classification model. It contains approximately 1.8 million images from 365 place categories and is a version of the Places365 dataset[1], a subset of the Places2 dataset (ZHOU et al., 2018). The dataset has indoor, nature, and urban as macro-classes. However, as the scope of this thesis is indoor environments and the following experiments are performed in home settings, we selected data from 6 place categories commonly present in home environments to train the places classification model. The categories were: bathroom, bedroom, corridor, home office, kitchen, and living room.

Each category of the Places365-Standard has 3068 to 5000 training images and 100 validation images. Therefore, we used 30000 (5000 each category) and 600 images (100 each category) in the places classifier training and validation processes, respectively. Figure 17 exemplifies images from the selected place categories.

Figure 17 – Example images of each selected place category from Places365-Standard dataset.

Source: Adapted from Zhou et al. (2018).

5.4.2.2 *COCO dataset*

For training the multi-label objects classifier we used the latest version of the COCO (Common Objects in Context) dataset (LIN et al., 2014), released in the year of 2017[2]. The dataset introduces everyday complex scenes of objects in their natural context and was designed for object detection, segmentation, person keypoints detection, and captioning. The 2017 version has 123287 images (118287 for training and 5000 for validation), 80 object categories and 886284 annotated object instances, with the presence of multiple instances and categories of objects per image being the usual (Figure 18 shows examples).

[1] URL: http://places2.csail.mit.edu/
[2] URL: https://cocodataset.org

We selected 10 object categories: chair, couch, bed, dining table, toilet, TV, laptop, oven, sink, and refrigerator. Then, we used the annotated objects instances to generate a 10-dimensional ground truth vector for each image in the dataset that contains any instance of the selected categories, where each dimension represents an object category which receives value 1 if an instance of the object is present in the image or 0 if not (example of a resulting vector: 1 1 0 1 0 1 0 0 0 0). Thus, ground truth vectors were generated for 34474 training images and 1497 validation images. Finally, these images and their respective ground truth vectors were used in the training and validations processes of the multi-label objects classifier.

Additionally, as the classification outputs obtained from the multi-label objects classifier are sigmoidal, thresholds are needed to define the presence (1) or not (0) of the object labels. Therefore, for each output dimension of the classifier, we performed threshold moving on the outputs obtained from the validation images, calculated the F1-score for each threshold variation, and selected the value that maximized the result. The 10 selected thresholds (one for each object label) were used in all of the experiments.

Figure 18 – Example images from COCO dataset: (a) shows an image with containing two couch instances and one TV instance; (b) shows an image containing multiple instances of chair, as well as single instances of oven, refrigerator and others.

(a) (b)

Source: Adapted from Lin et al. (2014).

5.4.2.3 SUN3D dataset

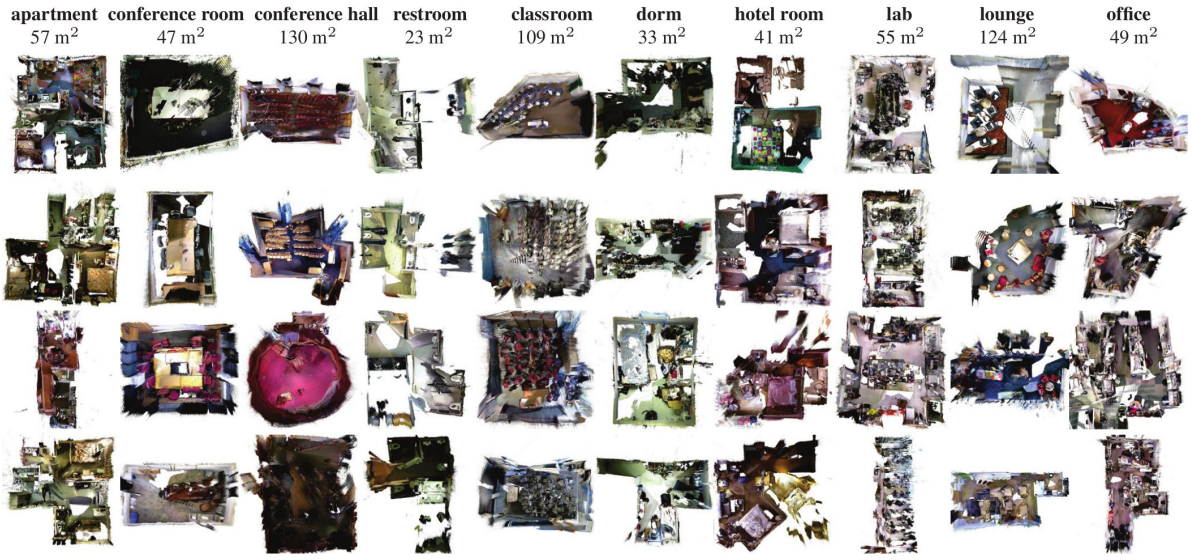
The experiments performed to evaluate the method use data selected from the SUN3D dataset (XIAO; OWENS; TORRALBA, 2013), a place-centric RGB-D dataset[3] that provides data sequences captured in the full 3D extent of many indoor places. The authors use an ASUS Xtion PRO LIVE sensor mounted to a laptop and capture the RGB-D sequences at a height and viewing angle similar to that of a person, i.e., a viewing angle that is mostly horizontal but slightly tilted towards the ground. The operators mimic human exploration in the capturing

[3] URL: https://sun3d.cs.princeton.edu/

process. They walk slowly through the entire space, scanning each room, including floor, walls and every object. The dataset has 415 data sequences captured from 254 different spaces, in 41 buildings mainly distributed across North America, Europe and Asia. Some places were scanned multiple times, on different days and times of the day.

The data sequences are from various types of environments of different sizes, such as apartment, conference hall, classroom, dorm, lab, lounge, office, and others (Figure 19 shows examples). However, for the experiments, we only selected data sequences from paths that represent complete environments (that is, environments with multiple categories of places) of houses and apartments. We selected one data sequence per path and the paths were: home_at, home_han, home_md, home_puigpunyen, and home_rz. From path home_md we used the data sequence 9 (out 10 available), as it covers the entire apartment, and from home_puigpunyen we used data sequence 2 (out 4 available), as it covers the entire house floor with the number of images close to the average of the other sequences. The other selected paths have only one data sequence available.

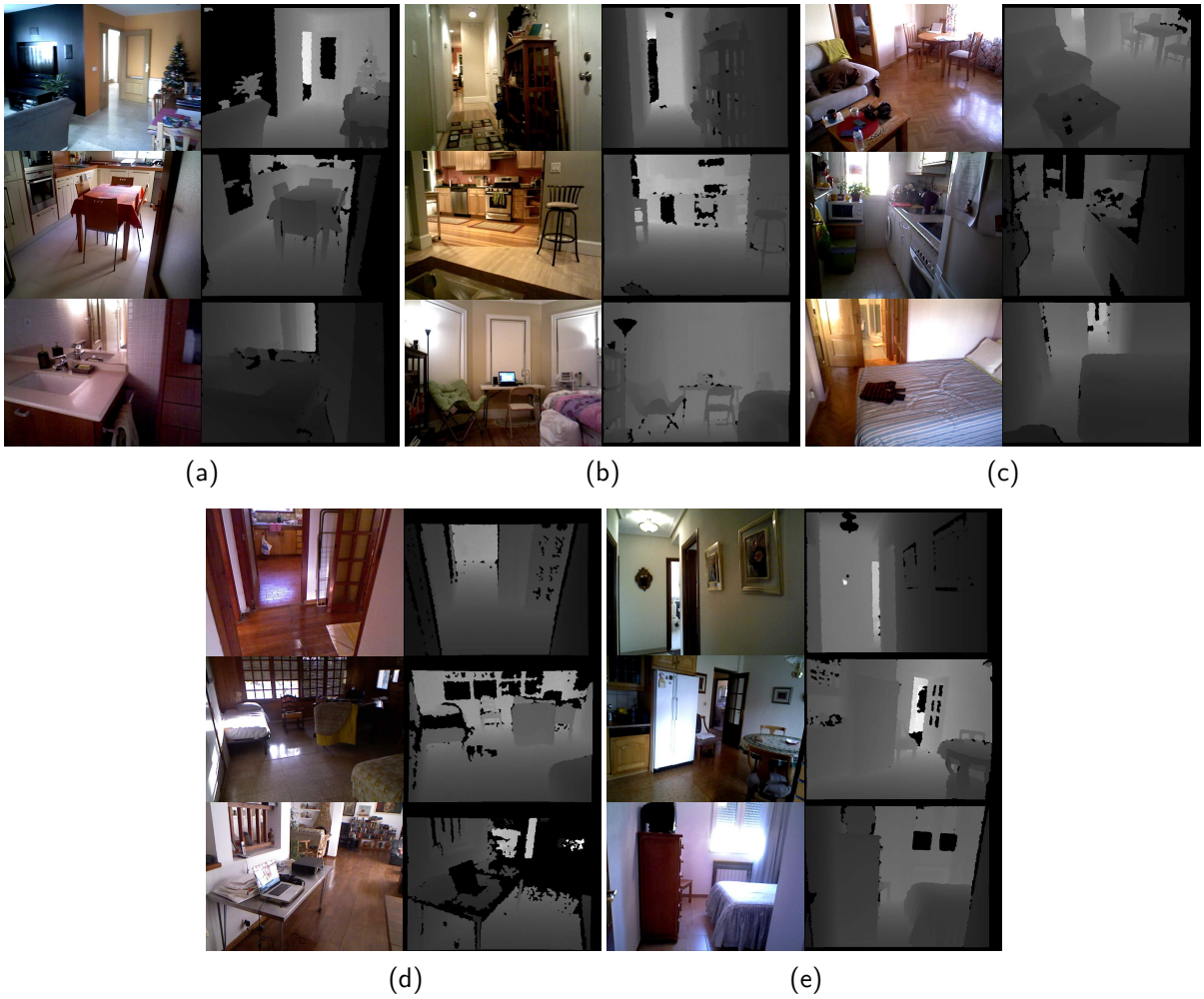
Figure 19 – In each column are examples of point clouds from different environment types and their median coverage areas.



Source: Xiao, Owens and Torralba (2013).

The environments covered by the selected paths vary in size and also in number of images, where: path home_at has 14785 RGB images and 14851 depth images; path home_han has 8595 RGB images and 8595 depth images; path home_md has 13341 RGB images and 13401 depth images; path home_puigpunyen has 16726 RGB images and 16726 depth images; and path home_rz has 30333 RGB images and 30469 depth images. Path home_rz is the one that covers the largest environment, an entire house. Paths home_han and home_md cover apartments. Paths home_at and home_puigpunyen cover entire floors of houses. Figure 20 shows some open-angle images of each selected path.

Figure 20 – Examples of RGB and Depth images of the selected paths from the SUN3D dataset: (a) home_at path; (b) home_han path; (c) home_md path; (d) home_puigpunyen path; and (e) home_rz path.



Source: Adapted from Xiao, Owens and Torralba (2013).

5.4.2.4 Data Preparation

The SUN3D dataset doesn't provide rosbags of the data sequences. A rosbag is a ROS file format for storing ROS communication data and is used to play back real-time ROS simulations (QUIGLEY et al., 2009). Therefore, to enable real-time experiments, we developed a ROS-based software tool[4] adapted for SUN3D data, which generates rosbags from sequences of image files (RGB and Depth), and used it to generate a rosbag for each of the selected data sequences.

The dataset also doesn't provide many ground truth semantic annotations, only for 8 data sequences from environment types unsuitable for our experiments. Thus, we annotated the RGB images of the data sequences from the selected paths with the 6 place categories defined in Section 5.4.2.1 and the 10 object categories defined in Section 5.4.2.2. The images captured

[4] Avaliable at: https://github.com/ygorsousa/rosbag_gen_utils

inside each room in each path received the most appropriate place category (out of the 6) and all objects present in the room (out of the 10 selected) as ground truth, i.e., each image received as ground truth a 10-dimensional vector (as described in Section 5.4.2.2) with the presence of all objects in the room, not those visualized in the image.

The annotations were made using part of the method in a real-time process, where for each selected data sequence: the corresponding rosbag provided the data in real-time; ORB-SLAM2 estimated the spatial position vector for each image; MobileNetV2 extracted the visual features vector from each image; then, the spatial position and visual features vectors corresponding to each image were stored in a file along with the respective ground truth data, provided in real-time through a software tool developed for the task.

Thus, a file was generated for each path[5] containing the following data for each image: id of the image; spatial position vector; visual features vector; place category (in numeric format); and 10-dimensional objects vector. In addition to the annotated ground truth, the spatial position and visual features vectors included allowed us to use the data in the process of hyperparameter adjustment for the experiments, described in Section 5.4.2.6.

The loss of some image frames during the annotation processes was expected since it was done in real-time. However, fortunately, a very small number of RGB frames was lost in most paths (6 in home_at, 2 in home_md, 1 in home_puigpunyen, and 42 in home_rz) and only in path home_han a larger number (558) was lost, which represents approximately 6.5% of the total number. This larger number was obtained because ORBSLAM2 lost some frames, probably due to the characteristics of the RGB images of this path, which are slightly dark and have a variable frame rate. Table 7 presents the number (by place category, by object category and total) of annotated images of each path.

5.4.2.5 Metric Maps

In order to maintain the stability of the metric spatial positions provided by the ORBSLAM2 over the executions and experiments, we first ran the ORBSLAM2 on each of the selected paths (using the rosbags) and saved the generated 3D metric maps. Then, for each of the previously described annotation processes, as well as in the experiments, ORBSLAM2 loaded the map corresponding to the considered path and acted only in localization mode. The ORBSLAM2 was always run in RGB-D mode, i.e., using both RGB and Depth images. Figure 21 shows a close-up view of the map generated for one of the selected paths and Figure 24 shows a top view of the metric maps generated for each of the selected paths.

[5] The generated files will be made available at: https://github.com/ygorsousa/semantic_annotations_sun3d

Table 7 – Number of annotated RGB images of each selected path from SUN3D. Numbers are presented by location category, object category, and total.

Place category	home_at	home_han	home_md	home_pui*	home_rz
Bathroom	2944	1163	2844	2461	3410
Bedroom	0	3085	2781	2754	5297
Corridor	1440	1044	0	1031	5568
Home office	0	0	0	0	3495
Kitchen	4302	1681	2193	3498	5610
Living room	6093	1064	5521	6981	6911
Object category					
Chair	10395	4149	5521	9735	21313
Couch	6093	1064	5521	6981	6911
Bed	0	3085	2781	2754	5297
Dining table	10395	0	5521	10479	5610
Toilet	2944	1163	2844	2461	3410
TV	10395	2377	5521	13233	19184
Laptop	0	1772	0	6981	0
Oven	4302	1681	2193	3498	5610
Sink	7246	2844	5037	5959	9020
Refrigerator	4302	1681	2193	3498	5610
Total number	14779	8037	13339	16725	30291

* home_pui stands for the home_puigpunyen.

Source: Thesis author.

5.4.2.6 Hyperparameter Adjustment

The process of hyperparameter adjustment for the experiments used the data annotated as described in Section 5.4.2.4 and followed a few steps. First, a single value for the spatial distance threshold λ was chosen to be used in all the experiments and in all variations of the method. Larger values for λ would decrease the number of nodes created in the topological maps, and, consequently, increase the results obtained only due to the smaller number of nodes. In contrast, smaller values would increase the number of nodes, and consequently could decrease the results obtained due to the lack of visual representativeness of the nodes.

Therefore, to find a suitable value for λ, we sampled all hyperparameters of the method (including ζ, introduced in the TCCMA process) within pre-established ranges using a uniform distribution and for each random configuration of hyperparameters, we calculated the average

Figure 21 – A close-up view of the feature-based 3D metric map generated for path home_rz using ORB-SLAM2.

Source: Thesis author.

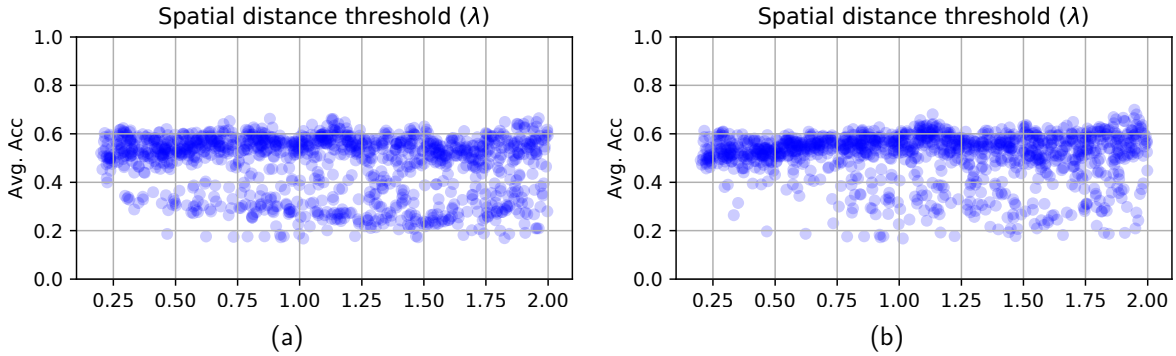
of the place classification accuracy results obtained in all selected paths, i.e., the average of the results obtained in all selected paths using the same hyperparameters setting. The averages were obtained with the method using each consolidation process (TCMA and TCCMA), separately, and an intermediate value in the sampled range that presented good average results for both processes was selected. Thus, the value chosen for λ was 1.1342 meters. Figure 22 illustrate in graphs the average accuracy results obtained by sampling the value of λ in the pre-established range.

Then, with the value of λ fixed, we sampled all the other hyperparameters within the same pre-established ranges and for each variation of the method (which includes both consolidation processes and the ablation variations) we selected the hyperparameter setting that presented the best average result of place classification (accuracy) obtained in all selected paths. The selected hyperparamarater settings are used in the classification of places and objects in the experiments (that is, the hyperparameter settings adjusted for the place classification results are also used for the multi-label objects classification) and are denoted as 'BAP' (Best in All Paths) in the result tables.

Next, from the same random values sampled for the hyperparemeters, we selected for each

variation of the method (including both consolidation processes and the ablation variations) the hyperparameter settings that presented the best place classification results in each path, i.e., a different hyperparameter setting for each path. The selected hyperparamarater settings are again used in the classification of places and objects in the experiments and are denoted as 'BEP' (Best in Each Path) in the result tables.

Figure 22 – Average accuracy results of places classification obtained by sampling the value of the spatial distance threshold λ in the pre-established range. (a) shows the results obtained with the method using the TCMA process and (b) the results obtained using the TCCMA process.



Source: Thesis author.

Lastly, in the hyperparameter adjustment process, all results were obtained using a version of the method made to operate with the annotated data. Therefore, the results obtained with the hyperparameter settings selected for both consolidation processes were validated in ROS simulations with the real-time version of the method. Table 8 presents the ranges used for each hyperparameter.

Table 8 – Hyperparameter ranges.

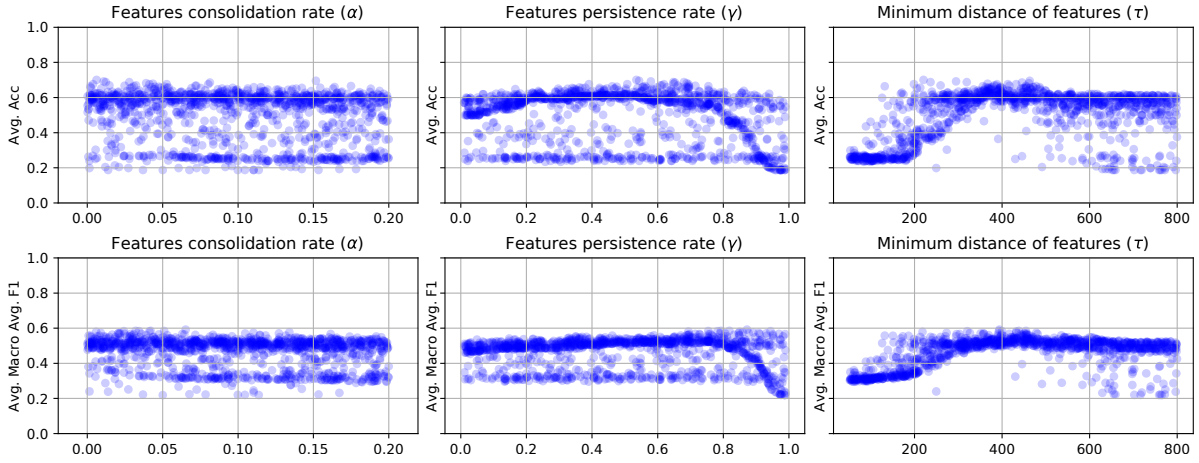
Hyperparameter	min	max
Features consolidation rate (α)	0.0001	0.2
Minimum distance of features (τ)	50.0	800.0
Features persistence rate (γ)	0.01	0.99
Distance threshold of consolidation items (ζ)	50.0	500.0

Source: Thesis author.

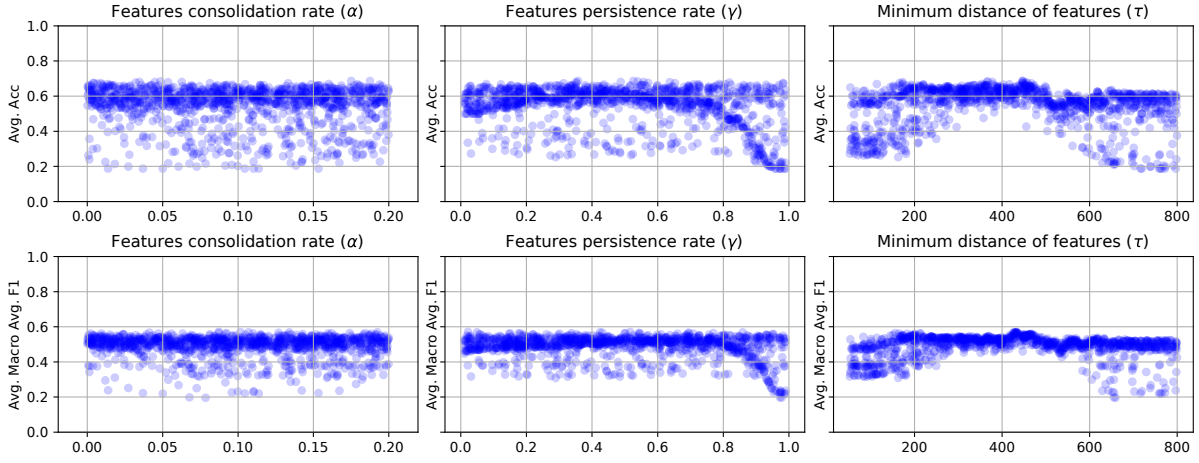
In addition, the hyperparameter that had the most effect on the average of the results obtained in each path (both in the classification of places and objects) with the two consolidation processes (TCMA and TCCMA) was the minimum distance of features τ, followed by the features persistence rate γ, as shown in the graphs illustrated in Figure 23. The hyperparameter τ controls the visual habituation of the nodes, which directly affects the number of consolidation updates of each node. The distance threshold of consolidation items ζ also

demonstrated having effect on the results with the TCCMA process, ζ controls the number of consolidation items created to represent the different patterns in the input features.

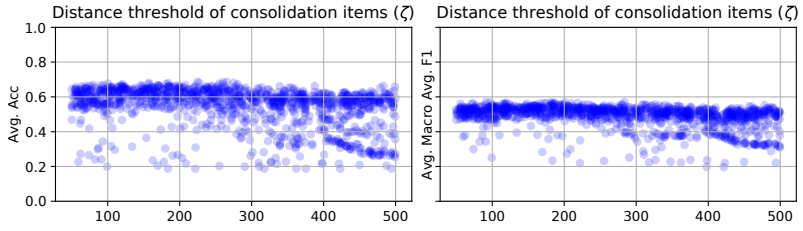
Figure 23 – Results of average accuracy of place classification and average Macro Average F1-score of objects classification obtained with all hyperparameters sampled within the pre-established ranges (as per Table 8), except spatial distance threshold λ that was fixed with the value of 1.1342 meters. (a) and (b) shows the results obtained with method using the TCMA process and the TCCMA process (respectively). (c) shows the results obtained with method using the TCCMA process for the hyperparameter ζ.



(a)



(b)



(c)

Source: Thesis author.

5.4.3 Topology

In the previous chapter (Section 4.2.3), the positioning of the nodes produced with the method was evaluated in relation to the two-dimensional spatial position data provided by the dataset used in the experiments. In addition, this section evaluates whether the positioning of the topological nodes produced is coherent with the three-dimensional spatial positions provided by ORBSLAM2. For this, a topological map was generated for each path selected from SUN3D, then the mean and maximum Euclidean distances between all position data and their nearest nodes were computed.

The results obtained were: 0.381 / 1.022 (mean / maximum distance) for home_at path; 0.396 / 1.084 (mean / maximum distance) for home_han path; 0.382 / 0.963 (mean / maximum distance) for home_md path; 0.440 / 1.081 (mean / maximum distance) for home_puigpunyen path; and, 0.423 / 1.023 (mean / maximum distance) for home_rz path. The results indicate that all nodes were positioned coherently with the 3D metric position data provided by ORBSLAM2 (with mean distance around 0.4 meters) and the spatial distance threshold λ (with maximum distance always below $\lambda = 1.1342$ meters). We also verified that all connections between nodes represented viable transitions in the environments. Figure 24 shows a topological map generated for each path.

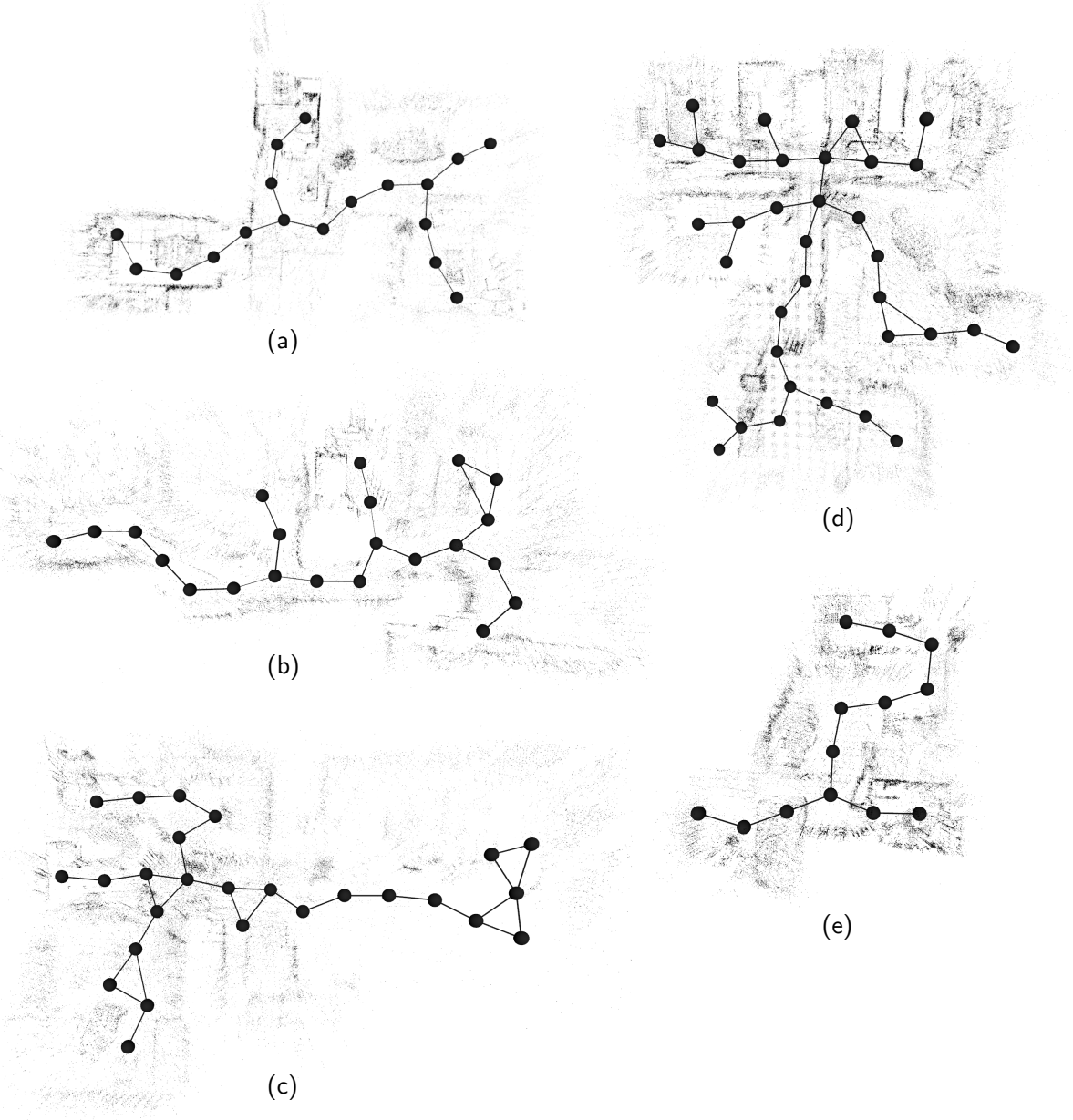
5.4.4 Place Classification

This section presents the first part of the experiments performed to evaluate the application of the method in more practical scenarios. The method is evaluated on the 5 selected paths using the two proposed consolidation processes (TCMA and TCCMA), separately. For each consolidation process, two hyperparameters scenarios are considered.

In the first scenario, the consolidated visual features are produced using the same hyperparameter setting for all paths (the 'BAP' hyperparameter setting described in Section 5.4.2.6). Then, the consolidated visual features vector $\mathbf{c}_j$ of each topological node j present in the generated maps is classified by the places classifier (introduced in Section 5.1.2), which indicates the place category of each node among the 6 available. In the second, the consolidated visual features are produced using the best hyperparameter setting obtained for each path (the 'BEP' hyperparameter settings described in Section 5.4.2.6). Again, the consolidated visual features vector $\mathbf{c}_j$ of each topological node j present in the generated maps is classified by the places classifier.

Table 9 presents the excellent results obtained, where the results with the best hyperparameters found for each path (BEP) are higher than all others and the results obtained with the same hyperparameters for all paths (BAP) are still very good, being mostly higher than the results obtained with the TC method (introduced in Section 5.3). Therefore, the results suggest that both consolidation processes (TCMA and TCCMA) generate adequate

Figure 24 – Top view of the topological maps generated using the method and their respective metric maps. In (a) are the maps for home_at path, (b) for home_han, (c) for home_puigpunyen, (d) for home_rz, and (e) for home_md.



Source: Thesis author.

visual representations (even if the hyperparameter values are not the best) to be used in the accurate recognition of place categories from an MLP classifier trained on data from other environments.

However, the places classifier seems to perform poorly in some categories of the selected paths, as demonstrated in the results obtained with the TC method (which uses the individual classification of images). The low performance in some classes may be related to the data used to train the classifier. Despite the large-scale dataset used, the training data may not represent the visual characteristics of the paths in some place categories. This probably affected the

classifications of the consolidated vectors in some classes, which decreased the overall accuracy results in some paths. Emphasis on the classification results of the living room category in path home_md, which were zero in all scenarios.

Table 9 also shows that the results obtained with the two consolidation processes are very similar in overall accuracy when using the best hyperparameters found for each path (BEP). However, the overall accuracy results obtained with the TCMA process were more balanced when using the same hyperparameters in all paths (BAP) than the results obtained with the TCCMA process, which indicates that the TCCMA process may be more difficult to tune for more general environments than the TCMA process.

Table 10 adds results obtained with variations of the consolidation processes without their visual habituation and visual persistence capabilities. Despite the accuracy of place classifications being a metric that is not very sensitive to detect small variations in the consolidated representations, the results indicate that both visual habituation and visual persistence present a positive contribution to the TCMA process. However, the results obtained with the TCCMA indicate that despite the visual habituation and visual persistence capabilities still having a positive contribution, they seem to have less significance in the consolidation process. This may be related to the use of multiple consolidation items per node, which may have helped to avoid tainting the resulting consolidation vectors with repeated visual information.

Additionally, Table 11 presents the average number of updates performed in the consolidated features vector of each node during the evaluations. As expected, the average number of updates is similar in all paths when using the same hyperparameter settings (BAP) and very different when using the best hyperparameter settings found for each path (BEP). The numbers also show that the TCMA process appears to perform better on average with more updates than the TCCMA process, but that the TCCMA process can also perform well with a high number of updates. It is important to note that the number of updates in each path is different even with the same hyperparameter setting, as the number of images and sizes of each path are different.

Table 11 also presents the average number of consolidation items created in each node during the evaluations with the TCCMA process. Interestingly, most of the numbers are very similar to the number of updates, which ends up causing the consolidated features vector of each node to be calculated almost as the simple average of the features vectors that each consolidation item was initialized, since the consolidation items had almost no updates. Evaluating this behavior is a future work.

5.4.5 Multi-Label Object Classification

This section presents the second part of the experiments. This part enriches all the maps produced in the previous section with a new semantic property, the objects present (among 10 available) in the regions covered by each topological node. For this, each node j present

Table 9 – Place classification results.

Accuracy	Bathroom	Bedroom	Corridor	Home office	Kitchen	Living room	Overall
$TCMA$[1]| BAP[2]							
home_at	1.000	—	0.667	—	1.000	0.750	0.833
home_han	1.000	1.000	0.714	—	1.000	0.500	0.773
home_md	1.000	1.000	—	—	1.000	0.000	0.615
home_pui[3]	1.000	1.000	0.500	—	1.000	0.333	0.654
home_rz	1.000	1.000	0.500	0.333	0.714	0.333	0.629
$TCMA$ | BEP							
home_at	1.000	—	1.000	—	1.000	1.000	1.000
home_han	1.000	1.000	0.857	—	1.000	0.500	0.818
home_md	1.000	1.000	—	—	1.000	0.000	0.615
home_pui	1.000	1.000	0.0	—	1.000	0.667	0.769
home_rz	1.000	1.000	0.500	0.667	0.714	0.444	0.686
$TCCMA$ | BAP							
home_at	1.000	—	1.000	—	1.000	0.875	0.944
home_han	1.000	0.800	0.571	—	1.000	0.167	0.591
home_md	1.000	1.000	—	—	0.500	0.000	0.538
home_pui	1.000	0.800	0.500	—	1.000	0.583	0.731
home_rz	1.000	1.000	0.500	0.000	0.714	0.444	0.629
$TCCMA$ | BEP							
home_at	1.000	—	1.000	—	1.000	1.000	1.000
home_han	1.000	1.000	1.000	—	1.000	0.333	0.818
home_md	1.000	1.000	—	—	1.000	0.000	0.615
home_pui	1.000	1.000	0.500	—	1.000	0.583	0.769
home_rz	1.000	1.000	0.667	0.000	1.000	0.444	0.714
TC							
home_at	1.000	—	1.000	—	1.000	0.500	0.778
home_han	1.000	0.800	0.714	—	0.500	0.000	0.545
home_md	1.000	1.000	—	—	1.000	0.000	0.615
home_pui	1.000	1.000	1.000	—	1.000	0.417	0.731
home_rz	1.000	0.625	0.833	0.000	0.286	0.111	0.429

[1] $TCMA$ is the method using the TCMA process, $TCCMA$ is the method using the TCCMA process, and *TC* is the modification of the method that performs topological counting of place categories recognized from individual images.

[2] BAP is the hyperparameter setting that presented the best average result in all paths and BEP is the best hyperparameter setting obtained for each path.

[3] home_pui stands for home_puigpunyen and — means that the category is not present in the path.

Source: Thesis author.

Table 10 – Place classification. Evaluation with variations of the consolidation processes.

Accuracy	Hyperparam. Setting	home_at	home_han	home_md	home_pui	home_rz
$TCMA^*$	BAP	0.833	0.773	0.615	0.654	0.629
$TCMA_{-VH}$	BAP	0.833	0.727	0.461	0.731	0.514
$TCMA_{-VP}$	BAP	0.833	0.636	0.538	0.577	0.571
$TCMA_{-VHVP}$	BAP	0.833	0.636	0.461	0.538	0.571
$TCMA$	BEP	1.000	0.818	0.615	0.769	0.686
$TCMA_{-VH}$	BEP	1.000	0.727	0.615	0.731	0.629
$TCMA_{-VP}$	BEP	0.889	0.727	0.538	0.731	0.657
$TCMA_{-VHVP}$	BEP	0.833	0.682	0.538	0.654	0.571
$TCCMA$	BAP	0.944	0.591	0.538	0.731	0.629
$TCCMA_{-VH}$	BAP	0.889	0.454	0.615	0.654	0.686
$TCCMA_{-VP}$	BAP	0.944	0.545	0.615	0.654	0.657
$TCCMA_{-VHVP}$	BAP	0.889	0.500	0.615	0.654	0.657
$TCCMA$	BEP	1.000	0.818	0.615	0.769	0.714
$TCCMA_{-VH}$	BEP	0.944	0.727	0.615	0.692	0.714
$TCCMA_{-VP}$	BEP	0.944	0.682	0.615	0.731	0.714
$TCCMA_{-VHVP}$	BEP	0.944	0.682	0.615	0.692	0.714

* $TCMA$ is the method using the TCMA process and $TCCMA$ is the method using the TCCMA process. $TCMA_{-VP}$ is the TCMA without Visual Persistence, $TCMA_{-VH}$ is the TCMA without Visual Habituation, and $TCMA_{-VHVP}$ is TCMA without Visual Habituation and Visual Persistence. The same nomenclature applies to the TCCMA process.

Source: Thesis author.

in all maps, including the maps generated with the two consolidation processes and both hyperparameter scenarios, has its consolidated visual features vector $\mathbf{c}_j$ classified by the multi-label objects classifier (introduced in Section 5.1.2), which indicates the presence or not of 10 different classes of objects.

In order to use the TC method (Section 5.3) for comparison in the experiments, we selected for each path a different object counts threshold in a range of 1 and 1000 that maximized the Macro Average F1-score obtained (Figure 25 illustrates the process in graphs). In the TC method, the objects count threshold determines which object labels are present in each node j from its objects presence counts vector $\mathbf{g}_j$.

Table 12 presents the results obtained, where the results of Micro Average F1-score obtained with the best hyperparameters found for each path (BEP) are in general a little higher (mainly those of TCMA) than the results obtained with the method TC and the results of Macro Average F1-score obtained (also with the BEP hyperparameter settings) are generally similar to those obtained with the TC method. Although the results obtained with the same hyperparameters for all paths (BAP) are not as good as the others, they are still good in some paths.

Table 11 – Number of updates and consolidation items (for $TCCMA$ only) per node. Standard deviations are shown in parentheses.

No. of Updates	home_at	home_han	home_md	home_pui	home_rz
$TCMA^*$| BAP	69.5 (41.6)	34.5 (29.4)	113.5 (57.2)	57.7 (23.7)	73.9 (36.4)
$TCMA$ | BEP	21.7 (14.6)	2.4 (3.1)	11.6 (5.2)	140.9 (55.5)	31.1 (15.3)
$TCCMA$ | BAP	6.1 (4.4)	2.5 (2.8)	10.8 (5.1)	4.5 (2.5)	7.5 (4.5)
$TCCMA$ | BEP	13.6 (9.4)	5.1 (4.3)	313.5 (154.3)	14.8 (6.6)	197.6 (99.1)
No. of Consolidation Items	home_at	home_han	home_md	home_pui	home_rz
$TCCMA$ | BAP	7.0 (4.4)	3.5 (2.7)	11.5 (5.0)	5.4 (2.3)	8.4 (4.4)
$TCCMA$ | BEP	13.9 (9.0)	1.5 (0.8)	87.4 (25.1)	15.7 (6.5)	79.7 (26.2)

* $TCMA$ is the method using the TCMA process and $TCCMA$ is the method using the TCCMA process.

Source: Thesis author.

Despite the fact that the hyperparameter settings used were adjusted for the place classification task, the multi-label objects classification results obtained with the hyperparameter settings 'BEP' were very good and comparable to those obtained with the TC method, which had the objects count threshold adjusted for each path. This suggests that new semantic properties can be extracted from the visual representations consolidated in previously created maps, even if the hyperparameter values used to generate the representations are not optimal for the new property.

Furthermore, it is important to note that the classifications are evaluated by the objects present in the room in which each node is positioned, and not by the objects visualized in the images consolidated in the node. This is a more difficult task because the images consolidated by many nodes may not capture some objects that are present in the rooms where they are positioned. Also, the classification threshold of each output dimension in the multi-label objects classifier was adjusted by the validation data selected from the COCO dataset, which may not have produced optimal thresholds to be used in environments with the different visual characteristics of the selected paths.

In addition to the results already discussed, Table 13 presents the results obtained with the hyperparameter settings that yielded the best multi-label objects classification results (Macro Average F1-score) in each path, which includes results with variations of both consolidation processes without their visual habituation and visual persistence capabilities. The results are additional evidence of the positive contribution of both visual habituation and visual persistence in the TCMA consolidation process. Furthermore, the results show that these capabilities also contribute positively to the TCCMA consolidation process, but reinforce that they have less significance.

Table 13 also shows that the results obtained with the TCMA process were mainly higher than the results obtained with the TCCMA process in agreement with the results obtained

Table 12 – Multi-label objects classification results.

F1-Score	Chair	Couch	Bed	Dining table	Toilet	TV	Laptop	Oven	Sink	Refrigerator	Micro Avg.	Macro Avg.	Macro Avg. B[4]
$TCMA$[1]| BAP[2]													
home_at	0.800	0.823	—	0.800	0.800	0.526	—	1.000	0.933	0.833	0.791	0.652	0.815
home_han	0.833	0.545	0.667	—	0.500	0.667	0.000	0.500	0.471	0.500	0.582	0.468	0.520
home_md	0.714	0.750	0.727	0.333	0.667	0.000	—	0.000	0.889	0.000	0.594	0.408	0.453
home_pui[3]	0.872	0.667	0.800	0.625	0.667	0.429	0.000	0.364	0.778	0.545	0.620	0.575	0.575
home_rz	0.906	0.762	0.750	0.600	0.333	0.160	—	0.857	0.621	0.727	0.660	0.572	0.635
$TCMA$ | BEP													
home_at	0.846	0.889	—	0.750	0.800	0.667	—	1.000	0.933	0.667	0.812	0.655	0.819
home_han	0.870	0.800	0.833	—	0.500	0.615	0.000	0.500	0.500	0.444	0.617	0.506	0.562
home_md	0.714	1.000	1.000	0.750	1.000	0.750	—	0.000	0.889	0.800	0.823	0.690	0.767
home_pui	0.809	0.647	0.889	0.686	0.800	0.667	0.000	0.600	0.750	0.714	0.676	0.656	0.656
home_rz	0.926	0.800	0.667	0.571	0.286	0.160	—	0.857	0.600	0.727	0.650	0.559	0.622
$TCCMA$ | BAP													
home_at	0.846	0.941	—	0.667	0.800	0.632	—	0.909	0.823	0.727	0.787	0.634	0.793
home_han	0.880	0.800	0.667	—	0.571	0.461	0.000	0.400	0.533	0.444	0.587	0.476	0.529
home_md	0.769	0.889	0.727	0.571	0.571	0.333	—	0.000	1.000	0.000	0.667	0.486	0.540
home_pui	0.820	0.687	0.889	0.621	0.667	0.370	0.154	0.800	0.778	0.727	0.649	0.651	0.651
home_rz	0.898	0.571	0.737	0.636	0.364	0.000	—	0.769	0.615	0.444	0.611	0.503	0.559
$TCCMA$ | BEP													
home_at	0.800	0.889	—	0.783	0.800	0.556	—	1.000	0.933	0.909	0.816	0.667	0.834
home_han	0.762	0.667	0.833	—	0.400	0.364	0.000	0.250	0.471	0.500	0.535	0.425	0.472
home_md	0.833	1.000	0.800	0.889	0.667	0.571	—	0.667	0.889	0.000	0.761	0.632	0.702
home_pui	0.800	0.562	0.889	0.667	0.667	0.483	0.000	0.833	0.778	0.833	0.650	0.651	0.651
home_rz	0.902	0.571	0.778	0.667	0.333	0.160	—	0.857	0.615	0.444	0.636	0.533	0.592
TC													
home_at	0.818	0.857	—	0.762	0.800	0.632	—	0.833	1.000	0.889	0.810	0.659	0.824
home_han	0.667	0.444	0.889	—	0.667	0.286	0.500	0.800	0.615	1.000	0.612	0.587	0.652
home_md	0.769	1.000	0.571	0.889	0.571	0.889	—	0.667	0.727	1.000	0.747	0.708	0.787
home_pui	0.800	0.600	0.889	0.606	0.500	0.686	0.286	0.833	0.609	0.714	0.661	0.652	0.652
home_rz	0.931	0.621	0.500	0.560	0.133	0.632	—	0.560	0.429	0.378	0.550	0.474	0.527

[1] $TCMA$ is the method using the TCMA process, $TCCMA$ is the method using the TCCMA process, and *TC* is the modification of the method that performs topological counting of the presence of objects recognized from individual images.

[2] BAP is the hyperparameter setting that presented the best average result in all paths and BEP is the best hyperparameter setting obtained for each path.

[3] home_pui stands for home_puigpunyen and — means that the category is not present in the path.

[4] Macro Avg. F1-Score B is calculated using only the categories present in the path.

Source: Thesis author.

with the 'BEP' hyperparameter settings presented in Table 12, where the results obtained with the TCMA process were also slightly higher. This suggests that the TCMA process produces consolidated visual features that are more representative and suitable for classifying multi-label objects when optimal values of hyperparameters are used.

Finally, despite the good results, the objects classifier had problems recognizing some object labels from the consolidated features vectors. This could have been caused by a challenge we faced while creating the consolidation processes: how to handle the repeated presentation of very similar images. Even with the visual habituation, both processes seem to be consolidating similar visual features from different images that end up suppressing representative characteristics previously consolidated. The introduction of multiple consolidation vectors per node (in the TCCMA process) was an attempt to build consolidations that are more representative and less susceptible to this problem, but, despite being promising, the results with the current version of the TCCMA process were not as good as expected.

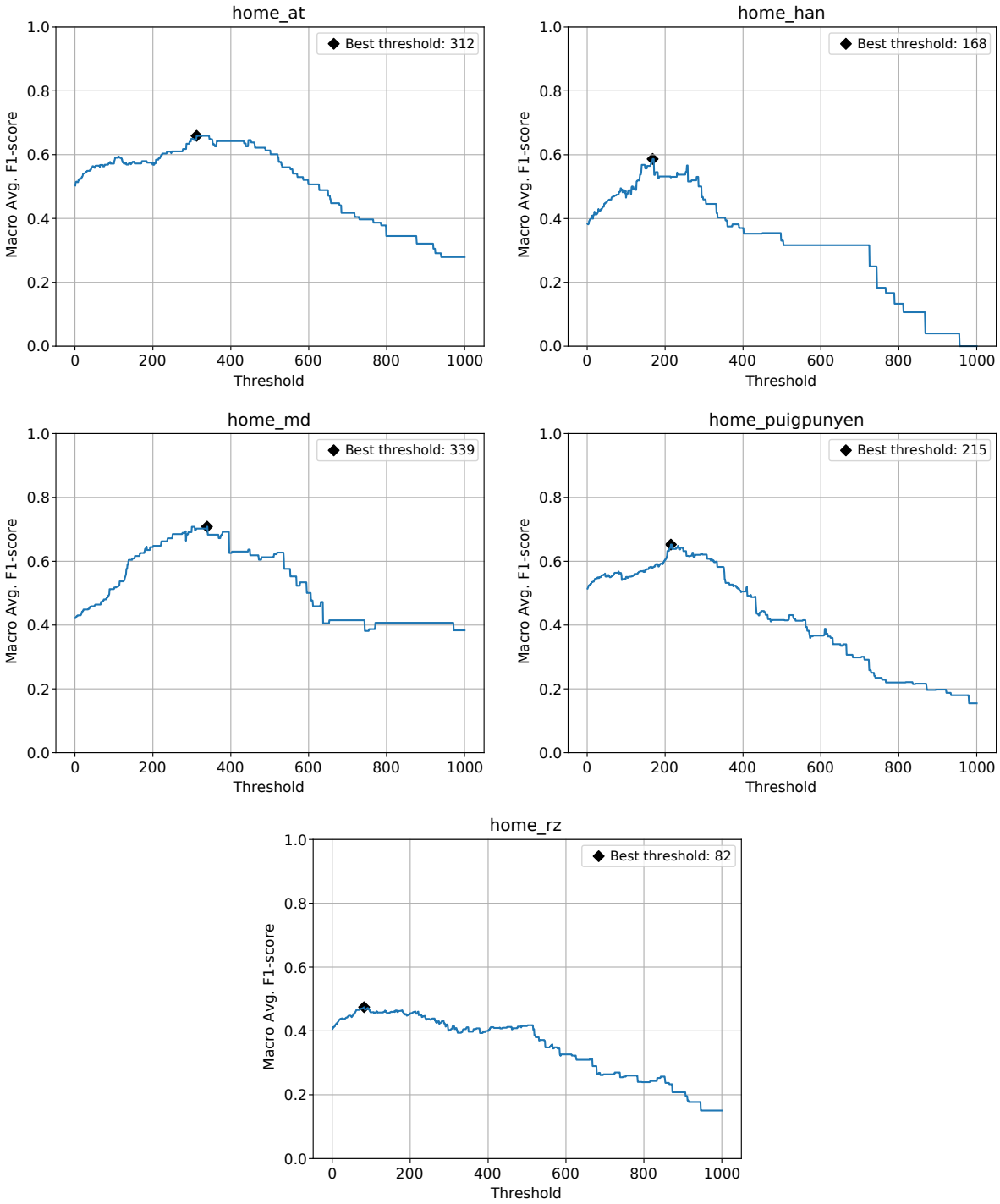
Table 13 – Multi-label objects classification. Results obtained with the hyperparameter settings that presented the best object classification results in each path.

Macro Avg. F1-Score B[1]	home_at	home_han	home_md	home_pui	home_rz
$TCMA$[2]	0.901	0.647	0.767	0.713	0.661
$TCMA_{-VH}$	0.896	0.575	0.637	0.697	0.638
$TCMA_{-VP}$	0.802	0.621	0.600	0.670	0.581
$TCMA_{-VHVP}$	0.803	0.555	0.544	0.631	0.579
$TCCMA$	0.884	0.654	0.736	0.699	0.636
$TCCMA_{-VH}$	0.845	0.575	0.757	0.685	0.610
$TCCMA_{-VP}$	0.849	0.625	0.736	0.683	0.625
$TCCMA_{-VHVP}$	0.831	0.575	0.757	0.652	0.610

[1] Macro Avg. F1-Score B is calculated using only the categories present in the path.
[2] $TCMA$ is the method using the TCMA process and $TCCMA$ is the method using the TCCMA process. $TCMA_{-VP}$ is the TCMA without Visual Persistence, $TCMA_{-VH}$ is the TCMA without Visual Habituation, and $TCMA_{-VHVP}$ is TCMA without Visual Habituation and Visual Persistence. The same nomenclature applies to the TCCMA process.

Source: Thesis author.

Figure 25 – The graphs show the Macro Avg. F1-score results obtained with the TC method for each path varying the objects count threshold between 1 and 1000. The best thresholds found are shown as black diamonds.



Source: Thesis author.

5.5 SUMMARY

This chapter presented an adaptation for real-time operation of the topological semantic mapping method proposed in Chapter 4. The adapted version was built as an integrated solution that aggregates a 3D metric SLAM method, a pre-trained CNN designed for resource-constrained environments, and two shallow MLP classifiers trained on large-scale datasets using the pre-trained CNN for transfer learning. In addition to the TCMA process, the first consolidation process proposed, a new visual features consolidation process denoted as Topological Consolidation of Features by the Combination of Multiple Moving Averages (TCCMA) was introduced.

The experiments, performed with data from 5 different home environments and in two scenarios of hyperparameter configuration, evaluated the method with the consolidated visual features produced with both TCMA and TCCMA processes, separately. The results suggested that the consolidated representations generated by both processes are suitable for accurate recognition of different semantic properties using classifiers trained on large-scale datasets of the literature (i.e, trained on data from other environments) and can be reused to adapt previously created maps for future applications through the recognition of new semantic properties.

The results also suggest that, despite not being as good as those obtained with the best hyperparameter values found for each environment, the consolidated representations produced with more general hyperparameter values are still rich and allow the recognition of different semantic properties.

Although the results of this research are very promising and bring significant progress to the generation of adaptive semantic maps by the use of consolidated representations of visual features, such innovation also uncovers additional challenges to be solved. The consolidation processes introduced in this thesis are just the beginning and numerous improvements can be considered. The next chapter describes some limitations identified in the developed solutions, proposes some directions for future work, and lists the contributions of this research.

6 CONCLUSIONS

This research investigated the creation of semantic maps that can automatically adapt their semantics even after the mapping process through the use of consolidated visual representations of spatial regions. First, a topological semantic mapping method that creates consolidated representations of the deep visual features extracted from RGB images captured around each topological node was proposed. In the method, the deep visual features are extracted using a pre-trained CNN and consolidated through the TCMA process, a deep visual features consolidation process that uses exponential moving averages and is empowered with visual persistence and visual habituation capabilities.

Then, a version for real-time operation of the topological semantic mapping method was introduced. The version was adapted into an integrated solution that aggregated a feature-based 3D metric SLAM method, a pre-trained CNN designed for resource-constrained systems, and two shallow MLP classifiers trained on large-scale datasets using the pre-trained CNN for transfer learning. In addition, the TCCMA process was introduced, a derivation of the TCMA process that combines multiple exponential moving averages.

The very promising results obtained with the proposed solutions presented significant advances in the objectives of this thesis. Experiments performed using an indoor real-world dataset suggested that the consolidated representations produced are rich visual representations of the topological regions they cover, fairly preserve the visual features of the consolidated images and do not degrade over time. The results also suggested that the consolidated representations are suitable for recognizing different semantic properties and for indicating the topological location of images. Furthermore, experiments performed in more practical scenarios indicated that the consolidated representations are suitable for recognizing semantic properties using classifiers trained on large-scale datasets from the literature and can be reused to adapt (i.e., modify or enhance) the semantics of previously created maps for new applications.

6.1 CONTRIBUTIONS

Given the above, the main contributions of this thesis are:

- A topological semantic mapping method that creates consolidated representations of deep visual features extracted from RGB images captured around each topological node and recognizes semantic information for the maps from the consolidated visual representations;
- A version of the method for real-time operation adapted into an integrated ROS-based solution that can be easily incorporated into real robots;

- A promising path for generating semantic maps that can automatically adapt their semantics through machine learning methods even after the mapping process;
- Strong evidence that exponential moving averages can be used to effectively consolidate deep visual features of spatial regions.

Other contributions are:

- Two processes of topological consolidation of deep visual features (the TCMA and TC-CMA) that use exponential moving averages as a basis and are endowed with mechanisms of visual persistence and visual habituation;
- Different experiments that evaluated variations of the methods in indoor environments and aimed to demonstrate the richness, stability and applicability of the consolidated visual features;
- Demonstration that the consolidated visual features are suitable for accurate recognition of different semantic properties and for other applications, such as topological location of images;
- The demonstration that the consolidated visual features can be reused to adapt the semantics of previously created maps to new requirements using classifiers trained on large-scale datasets found in the literature;
- The semantic annotation of the RGB images from 5 paths of the SUN3D dataset. The annotations included 6 place categories and 10 object categories.

During the development of this research, the method, experiments and results described in Chapter 4 were published in the IEEE Robotics and Automation Letters (2022). Below is the full citation of the paper:

- SOUSA, Y. C. N.; BASSANI, H. F. Topological semantic mapping by consolidation of deep visual features. IEEE Robotics and Automation Letters, v. 7, n. 2, p. 4110–4117, 2022.

6.2 LIMITATIONS

Despite the relevant progress obtained in this research, there are many limitations in the proposed solutions and evaluations performed. Some of them are discussed below:

- The CNNs used in the proposed methods were chosen because they are classic models in the literature. Their use was very important to demonstrate that the consolidated representations built are rich even with the original deep features extracted with CNNs that are not the most modern in the literature. However, the use of more recent CNN

models could provide richer visual features to the consolidation processes and improve the results obtained;

- The consolidation of deep visual features was evaluated only with the output of the adaptive average pooling layer of each CNN model used. The use of the output of other layers in the latent space still needs to be evaluated and could possibly lead to better results;
- There are scenarios in which the consolidation processes appear to be consolidating similar visual features from different images and end up suppressing representative characteristics already consolidated. This might be related to zigzag or circular visual movements, which would provide the repeated presentations of very similar images, or to the continuous consolidation of similar visual features from different angles and positions in the rooms. The visual habituation mechanism certainly needs improvements to better handle more advanced scenarios like these, however, further investigation is needed to better understand the causes of this behavior;
- The spatial area covered by each topological node has a spherical and fixed shape, which may not be suitable for different types of environment. Other coverage formats need to be evaluated, which may even be adaptive to the type of region considering other input data not used in this research so far, such as depth measures associated with the estimated metric position;
- Despite the promising results obtained with more general hyperparameter values (in the experiments of Chapter 5), the results obtained with the best hyperparameter values for each evaluated path were mostly superior. This indicates that the most appropriate hyperparameter values of the consolidation processes for different types of environments need to be studied further. Furthermore, this suggests that the proposed consolidation processes could be improved to better adapt to different environments. One possibility is to study a features consolidation rate adaptive to input patterns;
- Nodes positioned close to the borders between rooms end up consolidating specific visual characteristics of different rooms. This is probably the main cause that leads these nodes to be more frequently misclassified than the others in the experiments. This is a problem directly related to the use of visual information and the creation of solutions to mitigate it requires further study;
- The experiments performed did not evaluate the proposed solutions in dynamic environments. The use of exponential moving averages should give some robustness to the consolidation processes for this type of scenario since this type of average gives greater weight to the most recent data. However, the challenges in dynamic scenarios are many and a robust assessment would certainly provide evidence for improving the proposed solutions;

- The TCCMA process was evaluated only in the practical scenarios of experimentation. The use of multiple vectors of exponential moving averages seems promising to build consolidations that better represent different patterns in the input features, however, the consolidation process needs more detailed evaluations so that its behavior is better understood and improvements are proposed.

6.3 FUTURE WORK

The limitations described above lead to future steps that can be immediately explored to improve the solutions introduced in this research. Some of them are:

- Perform experiments in challenging scenarios to further evaluate the representativeness and stability of the consolidated visual representations over time;
- Evaluate the proposed solutions in other types of indoor environments, including static and dynamic;
- Perform a deeper analysis of the impact of each hyperparameter of the proposed solutions and indicate the most appropriate values for different types of indoor scenarios;
- Use more recent CNNs in the proposed methods and evaluate the consolidation of the output representations of layers in the latent space other than the adaptive average pooling layer;
- Extend the proposed consolidation processes to include a visual features consolidation rate adaptive to input patterns and more advanced mechanisms of visual habituation and visual persistence;
- Further study the use of multiple vectors of exponential moving averages in the consolidation processes;
- Study a technique that allows the adaptation of the spatial areas covered by each topological node to the shape of the covered region (e.g., up to the limit of a wall present in the covered region).

However, as this research investigated an innovative procedure for generating semantic maps, the solutions proposed and applications explored so far are just the starting point and many other possibilities for future directions can be considered from them, such as:

- Evaluate the use of the consolidated visual representations for the recognition of other less usual semantic information in semantic maps for indoor environments, such as activities normally performed in each room;

- Evaluate the use of the consolidated visual representations in applications that require the use of path planning. New semantic information integrated into the maps could allow immediate use and more efficient path planning;
- Expand the proposed solutions for creating multi-hierarchical maps with multiple levels of consolidated visual representations;
- Adapt the proposed solutions to outdoor environments. In urban scenarios, for example, the consolidated representations could be used to recognize different semantic information such as the type of area (e.g., commercial or residential) and if there is adequate artificial lighting;
- Investigate the literature to find new references that could improve the quality of the consolidated visual representations. One possibility is the literature on visual feature aggregation, present in different applications.

REFERENCES

ACHOUR, A.; AL-ASSAAD, H.; DUPUIS, Y.; ZAHER, M. E. Collaborative mobile robotics for semantic mapping: A survey. *Applied Sciences*, v. 12, n. 20, 2022. ISSN 2076-3417.

ANATI, R. C. *Semantic Localization and Mapping in Robot Vision*. Phd Thesis (PhD Thesis) — University of Pennsylvania, 2016.

ARAUJO, A. F.; REGO, R. L. Self-organizing maps with a time-varying structure. *ACM Comput. Surv.*, ACM, New York, NY, USA, v. 46, n. 1, Jul. 2013. ISSN 0360-0300.

ARAUJO, A. R.; COSTA, D. C. Local adaptive receptive field self-organizing map for image color segmentation. *Image and Vision Computing*, v. 27, n. 9, p. 1229 – 1239, 2009. ISSN 0262-8856.

BASSANI, H. *Modelos Neurais de Aquisição de Linguagem Natural para Agentes Incorporados*. Phd Thesis (PhD Thesis) — Universidade Federal de Pernambuco, 2014.

BASSANI, H. F.; ARAUJO, A. F. R. Dimension selective self-organizing maps for clustering high dimensional. In: *International Joint Conference on Neural Networks (IJCNN)*. [S.l.]: IEEE, 2012. ISSN 2161-4393.

BASSANI, H. F.; ARAUJO, A. F. R. Dimension selective self-organizing maps with time-varying structure for subspace and projected clustering. *IEEE Transactions on Neural Networks and Learning Systems*, v. 26, n. 3, p. 458–471, 2015.

BASTIANELLI, E.; BLOISI, D. D.; CAPOBIANCO, R.; COSSU, F.; GERMIGNANI, G.; IOCCHI, L.; NARDI, D. On-line semantic mapping. In: *International Conference on Advanced Robotics*. [S.l.]: IEEE Computer Society, 2013. p. 1–6.

BAY, H.; ESS, A.; TUYTELAARS, T.; VAN GOOL, L. Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, v. 110, n. 3, p. 346–359, 2008. ISSN 1077-3142.

BENGIO, Y.; COURVILLE, A.; VINCENT, P. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 35, n. 8, p. 1798–1828, 2013.

BERNUY, F.; SOLAR, J. Ruiz-del. Topological semantic mapping and localization in urban road scenarios. *Journal of Intelligent & Robotic Systems*, Springer, v. 92, n. 1, p. 19–32, 2018.

BRAGA, P. H. M.; BASSANI, H. F. A semi-supervised self-organizing map for clustering and classification. In: *2018 International Joint Conference on Neural Networks (IJCNN)*. [S.l.: s.n.], 2018. p. 1–8.

CADENA, C.; CARLONE, L.; CARRILLO, H.; LATIF, Y.; SCARAMUZZA, D.; NEIRA, J.; REID, I.; LEONARD, J. J. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, v. 32, n. 6, p. 1309–1332, 2016.

CHEN, W.; SHANG, G.; JI, A.; ZHOU, C.; WANG, X.; XU, C.; LI, Z.; HU, K. An overview on visual slam: From tradition to semantic. *Remote Sensing*, v. 14, n. 13, 2022. ISSN 2072-4292.

COLOMBO, J.; MITCHELL, D. W. Infant visual habituation. *Neurobiology of Learning and Memory*, v. 92, n. 2, p. 225–234, 2009. ISSN 1074-7427. Special Issue: Neurobiology of Habituation.

COLTHEART, M. Iconic memory and visible persistence. *Perception & psychophysics*, Springer, v. 27, p. 183–228, 1980.

DENG, J.; DONG, W.; SOCHER, R.; LI, L.-J.; LI, K.; FEI-FEI, L. ImageNet: A large-scale hierarchical image database. In: *IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2009. p. 248–255.

DUVALLET, F. *Natural Language Direction Following for Robots in Unstructured Unknown Environments*. Phd Thesis (PhD Thesis) — Carnegie Mellon University, 2015.

FANTZ, R. L. Visual experience in infants: Decreased attention to familiar patterns relative to novel ones. *Science*, v. 146, n. 3644, p. 668–670, 1964.

FELLBAUM, C. *WordNet: An electronic lexical database*. [S.l.]: MIT press, 1998.

FRITZKE, B. Growing cell structures—a self-organizing network for unsupervised and supervised learning. *Neural Networks*, v. 7, n. 9, p. 1441 – 1460, 1994. ISSN 0893-6080.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016.

GRINVALD, M.; FURRER, F.; NOVKOVIC, T.; CHUNG, J. J.; CADENA, C.; SIEGWART, R.; NIETO, J. Volumetric instance-aware semantic mapping and 3d object discovery. *IEEE Robotics and Automation Letters*, v. 4, n. 3, p. 3037–3044, 2019.

GU, J.; WANG, Z.; KUEN, J.; MA, L.; SHAHROUDY, A.; SHUAI, B.; LIU, T.; WANG, X.; WANG, G.; CAI, J.; CHEN, T. Recent advances in convolutional neural networks. *Pattern Recognition*, v. 77, p. 354–377, 2018. ISSN 0031-3203.

HAN, X.; LI, S.; WANG, X.; ZHOU, W. Semantic mapping for mobile robots in indoor scenes: A survey. *Information*, v. 12, n. 2, 2021. ISSN 2078-2489.

HAYKIN, S. *Neural Networks: A Comprehensive Foundation*. [S.l.]: Prentice Hall, 1998.

HELTON, J. C.; DAVIS, F. J.; JOHNSON, J. D. A comparison of uncertainty and sensitivity analysis results obtained with random and latin hypercube sampling. *Reliability Engineering and System Safety*, v. 89, p. 305–330, 2005.

HEMACHANDRA, S.; WALTER, M. R.; TELLEX, S.; TELLER, S. Learning spatial-semantic representations from natural language descriptions and scene classifications. In: *International Conference on Robotics and Automation*. [S.l.: s.n.], 2014. p. 2623–2630. ISSN 1050-4729.

HOWARD, A. G.; ZHU, M.; CHEN, B.; KALENICHENKO, D.; WANG, W.; WEYAND, T.; ANDREETTO, M.; ADAM, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

KOHONEN, T. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, v. 43, n. 1, p. 59–69, 1982. ISSN 1432-0770.

KOHONEN, T. The self-organizing map. *Proceedings of the IEEE*, v. 78, n. 9, p. 1464–1480, 1990.

KOLLAR, T.; PERERA, V.; NARDI, D.; VELOSO, M. Learning environmental knowledge from task-based human-robot dialog. In: *International Conference on Robotics and Automation*. [S.l.: s.n.], 2013. p. 4304–4309. ISSN 1050-4729.

KOSTAVELIS, I.; GASTERATOS, A. Semantic mapping for mobile robotics tasks: A survey. *Robotics and Autonomous Systems*, v. 66, p. 86 – 103, 2015. ISSN 0921-8890.

KUNZE, M.; STEFFENS, J. Growing cell structure and neural gas - incremental neural networks. In: *Proceedings of the Fourth AIHEP Workshop*. [S.l.]: World Scientific, 1995.

LAURITZEN, S.; RICHARDSON, T. Chain graph models and their causal interpretations. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, Blackwell Publishers, v. 64, n. 3, p. 321–348, 2002. ISSN 1467-9868.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *Nature*, v. 521, n. 7553, p. 436–444, 2015.

LI, Z.; LIU, F.; YANG, W.; PENG, S.; ZHOU, J. A survey of convolutional neural networks: Analysis, applications, and prospects. *IEEE Transactions on Neural Networks and Learning Systems*, v. 33, n. 12, p. 6999–7019, 2022.

LIN, T.-Y.; MAIRE, M.; BELONGIE, S.; HAYS, J.; PERONA, P.; RAMANAN, D.; DOLLÁR, P.; ZITNICK, C. L. Microsoft COCO: Common objects in context. In: *Computer Vision – ECCV 2014*. [S.l.]: Springer International Publishing, 2014. p. 740–755.

LOWE, D. Object recognition from local scale-invariant features. In: *Proceedings of the Seventh IEEE International Conference on Computer Vision*. [S.l.: s.n.], 1999. v. 2, p. 1150–1157 vol.2.

LUO, R. C.; CHIOU, M. Hierarchical semantic mapping using convolutional neural networks for intelligent service robotics. *IEEE Access*, v. 6, p. 61287–61294, 2018.

MA, L.; STUCKLER, J.; KERL, C.; CREMERS, D. Multi-view deep learning for consistent semantic mapping with RGB-D cameras. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. [S.l.: s.n.], 2017. p. 598–605.

MANCINI, M.; BULO, S. R.; RICCI, E.; CAPUTO, B. Learning deep NBNN representations for robust place categorization. *IEEE Robotics and Automation Letters*, v. 2, n. 3, p. 1794–1801, 2017.

MARSLAND, S.; SHAPIRO, J.; NEHMZOW, U. A self-organising network that grows when required. *Neural Networks*, v. 15, n. 8–9, p. 1041 – 1058, 2002. ISSN 0893-6080.

MATURANA, D.; ARORA, S.; SCHERER, S. Looking forward: A semantic mapping system for scouting with micro-aerial vehicles. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. [S.l.: s.n.], 2017. p. 6691–6698.

MATURANA, D.; CHOU, P.-W.; UENOYAMA, M.; SCHERER, S. Real-time semantic mapping for autonomous off-road navigation. In: HUTTER, M.; SIEGWART, R. (Ed.). *Field and Service Robotics*. [S.l.]: Springer International Publishing, 2018. p. 335–350. ISBN 978-3-319-67361-5.

MCCORMAC, J.; HANDA, A.; DAVISON, A.; LEUTENEGGER, S. Semanticfusion: Dense 3D semantic mapping with convolutional neural networks. In: *IEEE International Conference on Robotics and Automation (ICRA)*. [S.l.: s.n.], 2017. p. 4628–4635.

MUR-ARTAL, R.; MONTIEL, J. M. M.; TARDOS, J. D. Orb-slam: A versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, v. 31, n. 5, p. 1147–1163, 2015.

MUR-ARTAL, R.; TARDOS, J. D. ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras. *IEEE Transactions on Robotics*, v. 33, n. 5, p. 1255–1262, 2017.

NAKAJIMA, Y.; SAITO, H. Efficient object-oriented semantic mapping with object detector. *IEEE Access*, v. 7, p. 3206–3213, 2019.

NAKAJIMA, Y.; TATENO, K.; TOMBARI, F.; SAITO, H. Fast and accurate semantic mapping through geometric-based incremental segmentation. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. [S.l.: s.n.], 2018. p. 385–392.

NEWCOMBE, R. A.; IZADI, S.; HILLIGES, O.; MOLYNEAUX, D.; KIM, D.; DAVISON, A. J.; KOHI, P.; SHOTTON, J.; HODGES, S.; FITZGIBBON, A. Kinectfusion: Real-time dense surface mapping and tracking. In: *2011 10th IEEE International Symposium on Mixed and Augmented Reality*. [S.l.: s.n.], 2011. p. 127–136.

NEYSHABUR, B.; SEDGHI, H.; ZHANG, C. What is being transferred in transfer learning? In: LAROCHELLE, H.; RANZATO, M.; HADSELL, R.; BALCAN, M.; LIN, H. (Ed.). *Advances in Neural Information Processing Systems*. [S.l.]: Curran Associates, Inc., 2020. v. 33, p. 512–523.

NIETO, J.; GUIVANT, J.; NEBOT, E. Denseslam: Simultaneous localization and dense mapping. *The International Journal of Robotics Research*, v. 25, n. 8, p. 711–744, 2006.

OH, J.; SUPPE, A.; DUVALLET, F.; BOULARIAS, A.; VINOKUROV, J.; NAVARRO-SERMENT, L.; ROMERO, O.; DEAN, R.; LEBIERE, C.; HEBERT, M.; STENTZ, A. Toward mobile robots reasoning like humans. In: *AAAI Conference on Artificial Intelligence*. [S.l.: s.n.], 2015.

PASZKE, A.; GROSS, S.; MASSA, F.; LERER, A.; BRADBURY, J.; CHANAN, G.; KILLEEN, T.; LIN, Z.; GIMELSHEIN, N.; ANTIGA, L.; DESMAISON, A.; KOPF, A.; YANG, E.; DEVITO, Z.; RAISON, M.; TEJANI, A.; CHILAMKURTHY, S.; STEINER, B.; FANG, L.; BAI, J.; CHINTALA, S. Pytorch: An imperative style, high-performance deep learning library. In: *Advances in Neural Information Processing Systems 32*. [S.l.: s.n.], 2019. p. 8024–8035.

PAZ, D.; ZHANG, H.; LI, Q.; XIANG, H.; CHRISTENSEN, H. I. Probabilistic semantic mapping for urban autonomous driving applications. In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. [S.l.: s.n.], 2020. p. 2059–2064.

PRONOBIS, A. *Semantic Mapping with Mobile Robots*. Phd Thesis (PhD Thesis) — KTH Royal Institute of Technology, 2011.

PRONOBIS, A.; CAPUTO, B. COLD: The COsy localization database. *The International Journal of Robotics Research (IJRR)*, v. 28, n. 5, 2009.

PRONOBIS, A.; JENSFELT, P. Large-scale semantic mapping and reasoning with heterogeneous modalities. In: *International Conference on Robotics and Automation (ICRA)*. [S.l.]: IEEE, 2012. p. 3515–3522. ISSN 1050-4729.

QUIGLEY, M.; CONLEY, K.; GERKEY, B.; FAUST, J.; FOOTE, T.; LEIBS, J.; WHEELER, R.; NG, A. Y. ROS: an open-source robot operating system. In: KOBE, JAPAN. *ICRA (International Conference on Robotics and Automation) workshop on open source software*. [S.l.], 2009. v. 3, n. 3.2, p. 5.

RANGEL, J. C.; CAZORLA, M.; GARCÍA-VAREA, I.; ROMERO-GONZÁLEZ, C.; MARTÍNEZ-GÓMEZ, J. Automatic semantic maps generation from lexical annotations. *Autonomous Robots*, Springer, v. 43, n. 3, p. 697–712, 2019.

RODDICK, T.; CIPOLLA, R. Predicting semantic map representations from images using pyramid occupancy networks. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2020.

ROSINOL, A.; VIOLETTE, A.; ABATE, M.; HUGHES, N.; CHANG, Y.; SHI, J.; GUPTA, A.; CARLONE, L. Kimera: From SLAM to spatial perception with 3D dynamic scene graphs. *The International Journal of Robotics Research*, v. 40, n. 12-14, p. 1510–1546, 2021.

RUBIO, F.; MARTINEZ-GOMEZ, J.; JULIA FLORES, M.; PUERTA, J. M. Comparison between bayesian network classifiers and SVMs for semantic localization. *Expert Systems with Applications*, 2016. ISSN 0957-4174.

RUBLEE, E.; RABAUD, V.; KONOLIGE, K.; BRADSKI, G. Orb: An efficient alternative to sift or surf. In: *2011 International Conference on Computer Vision*. [S.l.: s.n.], 2011. p. 2564–2571.

SANDLER, M.; HOWARD, A.; ZHU, M.; ZHMOGINOV, A.; CHEN, L.-C. MobileNetV2: Inverted residuals and linear bottlenecks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2018.

SOUSA, Y. C. N.; BASSANI, H. F. Incremental semantic mapping with unsupervised on-line learning. In: *International Joint Conference on Neural Networks (IJCNN)*. [S.l.: s.n.], 2018.

SOUSA, Y. C. N.; BASSANI, H. F. Topological semantic mapping by consolidation of deep visual features. *IEEE Robotics and Automation Letters*, v. 7, n. 2, p. 4110–4117, 2022.

SUNDERHAUF, N.; DAYOUB, F.; MCMAHON, S.; TALBOT, B.; SCHULTZ, R.; CORKE, P.; WYETH, G.; UPCROFT, B.; MILFORD, M. Place categorization and semantic mapping on a mobile robot. In: *International Conference on Robotics and Automation (ICRA)*. [S.l.: s.n.], 2016. p. 5729–5736.

SUNDERHAUF, N.; PHAM, T. T.; LATIF, Y.; MILFORD, M.; REID, I. Meaningful maps with object-oriented semantic mapping. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. [S.l.: s.n.], 2017. p. 5079–5085.

SZEGEDY, C.; LIU, W.; JIA, Y.; SERMANET, P.; REED, S.; ANGUELOV, D.; ERHAN, D.; VANHOUCKE, V.; RABINOVICH, A. Going deeper with convolutions. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2015.

THRUN, S. Robotic mapping: A survey. *Exploring artificial intelligence in the new millennium*, v. 1, n. 1-35, p. 1, 2002.

WALTER, M. R.; HEMACHANDRA, S.; HOMBERG, B.; TELLEX, S.; TELLER, S. A framework for learning semantic maps from grounded natural language descriptions. *The International Journal of Robotics Research*, v. 33, n. 9, p. 1167–1190, 2014.

XIANG, Y.; FOX, D. DA-RNN: Semantic mapping with data associated recurrent neural networks. In: *Robotics: Science and Systems*. Cambridge, Massachusetts: [s.n.], 2017.

XIAO, J.; OWENS, A.; TORRALBA, A. SUN3D: A database of big spaces reconstructed using sfm and object labels. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. [S.l.: s.n.], 2013.

YOSINSKI, J.; CLUNE, J.; BENGIO, Y.; LIPSON, H. How transferable are features in deep neural networks? In: GHAHRAMANI, Z.; WELLING, M.; CORTES, C.; LAWRENCE, N.; WEINBERGER, K. (Ed.). *Advances in Neural Information Processing Systems*. [S.l.]: Curran Associates, Inc., 2014. v. 27.

ZHANG, Y.; WU, Y.; TONG, K.; CHEN, H.; YUAN, Y. Review of visual simultaneous localization and mapping based on deep learning. *Remote Sensing*, v. 15, n. 11, 2023. ISSN 2072-4292.

ZHOU, B.; LAPEDRIZA, A.; KHOSLA, A.; OLIVA, A.; TORRALBA, A. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 40, n. 6, p. 1452–1464, 2018.