

ON MODELING CONTEXT FROM OBJECTS
WITH A LONG SHORT-TERM MEMORY FOR
INDOOR SCENE RECOGNITION

CAMILA LARANJEIRA DA SILVA

ON MODELING CONTEXT FROM OBJECTS
WITH A LONG SHORT-TERM MEMORY FOR
INDOOR SCENE RECOGNITION

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação address do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação address.

ORIENTADOR: ERICKSON RANGEL DO NASCIMENTO
COORIENTADOR: ANÍSIO MENDES LACERDA

Maio de 2019

CAMILA LARANJEIRA DA SILVA

ON MODELING CONTEXT FROM OBJECTS
WITH A LONG SHORT-TERM MEMORY FOR
INDOOR SCENE RECOGNITION

Dissertation presented to the Graduate Program in Computer Science of the Federal University of Minas Gerais in partial fulfillment of the requirements for the degree of Master in Computer Science.

ADVISOR: ERICKSON RANGEL DO NASCIMENTO
CO-ADVISOR: ANÍSIO MENDES LACERDA

May 2019

Silva, Camila Laranjeira da

S586m On modeling context from objects with a long short-term memory for indoor scene recognition [manuscrito] / Camila Laranjeira da Silva. – 2019.
xiiv, 66 f. il.

Orientador: Erickson Rangel do Nascimento.

Coorientador: Anísio Mendes Lacerda.

Dissertação (mestrado) - Universidade Federal de Minas Gerais, Instituto de Ciências Exatas, Departamento de Ciência da Computação.

Referências: f.61-66.

1. Computação – Teses. 2 Inteligência artificial – Teses. 3. Reconhecimento de imagens – Teses. 4. Redes neurais (Computação) – Teses. I. Nascimento, Erickson Rangel do. II. Lacerda, Anísio Mendes. III. Universidade Federal de Ciência da Computação. III. Título.

CDU 519.6*82(043)




UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO


FOLHA DE APROVAÇÃO

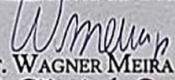
On Modeling Context from Objects with a Long Short-Term Memory for
Indoor Scene Recognition

CAMILA LARANJEIRA DA SILVA

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:


PROF. ERICKSON RANGEL DO NASCIMENTO - Orientador
Departamento de Ciência da Computação - UFMG


PROF. ANISIO MENDES LACERDA - Coorientador
Departamento de Ciência da Computação - UFMG


PROF. WAGNER MEIRA JÚNIOR
Departamento de Ciência da Computação - UFMG


DR. RENATO JOSÉ MARTINS
Pós-Doutorado - INRIA

Belo Horizonte, 28 de Maio de 2019.

Sou grata a todos que me mantiveram de pé, em vida ou em memória.

Acknowledgments

Se você está lendo essa página é porque eu consegui. Mas esse mérito está longe de ser exclusivamente meu, não teria chegado aqui sem as pessoas valiosas que já estavam e que apareceram na minha vida. Minha mãe, a melhor amiga que alguém poderia querer. Se tem alguém que me compreende e me aceita nesse mundo é ela, e mesmo tão longe se fez presente durante todo esse tempo. Meu pai, que em vida me construiu para ser forte, para não ter medo de um bom obstáculo. Hoje em memória ocupa um espaço cativo no meu coração, me lembrando sempre que eu consigo, eu posso, e eu mereço. Meu irmão Diego, que sempre me olhou com tanto orgulho. Queria conseguir ver o mundo com os olhos dele. E toda a minha família, de sangue ou de coração (é você mesmo Sílvia), cuja energia e alegria de viver fizeram uma falta fenomenal.

Um presente que recebi do mestrado foi a minha mais recente família. Aqui conheci a Vivi, minha companheira na profissão e na vida, sempre mantendo meus pés no chão e a minha cabeça no lugar. Meus irmãos de coração, Alan e Hugo, um há quase 10 anos compartilhando comigo alguns dos melhores momentos de nossas vidas, e o outro que mal chegou e já tem lugar no meu coração. Ambos foram essenciais para a construção deste trabalho, contribuindo com conhecimento técnico e apoio emocional. Todos os amigos que fiz no VeRLab que aguentaram minhas constantes conversas e pedidos de ajuda. Vocês fizeram eu me sentir em casa e acrescentaram muito no meu crescimento acadêmico. Os amigos do BahiaRT, alunos e professores, que sempre me apoiaram de longe. Vocês me ajudaram a ser mais aberta e a abraçar quem eu realmente sou. Durante cinco belos anos ajudaram a iniciar minha carreira acadêmica, devo muito a vocês.

Sou grata aos meus orientadores. O Erickson, um pesquisador fantástico, principalmente nos quesitos técnica e dedicação. O Anisio, cujo conhecimento foi de grande valia para a produção deste trabalho. E por fim, agradeço ao PPGCC e seus funcionários pelo suporte constante, e às agências de fomento CNPq, FAPEMIG e CAPES pelo fomento financeiro, que possibilitou a minha maior dedicação ao projeto.

*“And suddenly, just like that, hope became knowledge. I was going to win. It was just
a matter of when.”*
(Khaled Hosseini, *The Kite Runner*)

Resumo

O reconhecimento automático de cenas ainda é encarado como um desafio aberto na literatura, apesar de alguns trabalhos reportarem métricas de performance superior às dos seres humanos. Isso é especialmente válido para ambientes internos visto que eles podem ser bem representados pelos seus objetos, cuja variabilidade é muito alta. Objetos variam em ângulo, tamanho, textura, além de oclusões serem mais frequentes em cenas com muitos objetos. Apesar das Redes Neurais Convolutionais apresentarem uma performance excepcional para a maioria de problemas relacionados a imagens, para ambientes internos as melhores performances são atribuídas a abordagens que adicionam informação a nível de objeto, modelando a correlação entre eles. Sabendo que Redes Neurais Recorrentes foram projetadas para modelar a estrutura de uma dada sequência, recentemente surgiram pesquisas explorando suas vantagens aplicadas ao problema de reconhecimento de cenas. Apesar desses trabalhos comumente apresentarem resultados inferiores ao estado da arte, ainda há muito espaço para desvendar o potencial total de metodologias recorrentes. Portanto, este trabalho propõe representar uma imagem como uma sequência de partes de objeto, extraindo características semânticas de modelos pré treinados em grandes datasets de objetos, afim de alimentar uma rede Long Short-Term Memory bidirecional treinada para classificação de cenas. Nossa proposta de treinamento baseia-se na abordagem Muitos-Para-Muitos, tal que cada entrada possui uma predição de cena correspondente, permitindo o uso de cada predição individual para aumentar a qualidade da classificação através de uma votação ponderada das saídas. Nossa representação em forma de sequência, bem como a fusão de predições ao final ainda é pouco explorada por métodos da literatura baseado em abordagens recorrentes para reconhecimento de cenas. Nossa proposta foi avaliada em três datasets: Scene15, MIT67 e SUN397, superando o desempenho de todas as metodologias recorrentes no MIT67, um dataset completamente dedicado ao problema de ambientes internos. Enquanto os outros datasets, que misturam ambientes internos e externos, apresentaram um desafio maior para a nossa abordagem. No entanto, nós aprimoramos a performance em todos os datasets sobre os métodos mais bem sucedidos da literatura, pareando o nosso método com cada um deles através da composição de um ensemble de classificadores. Em outras palavras, uma estratégia conjunta com o nosso método se mostrou benéfica para a tarefa de reconhecimento de cenas.

Palavras-chave: Computação, Inteligência artificial, Reconhecimento de imagens, Redes neurais.

Abstract

Automatic scene recognition is still regarded as an open challenge, even though there are reports of outperforming human accuracy. This is specially true for indoor scenes, since they can be well represented by their composing objects, which is highly variable information. Objects vary in angle, size, texture, besides being often partially occluded on crowded scenes. Even though Convolutional Neural Networks showed remarkable performance for most image-related problems, for indoor scenes the top performances were attributed to approaches that added object-level information to the methodology, modeling their intricate relationship. Knowing that Recurrent Neural Networks were designed to model structure from a given sequence of elements, only recently researchers started exploiting its advantages applied to the problem of scene recognition. Even though such works are usually below the state of the art performance, there is still plenty of room to unravel the full potential of recurrent methodologies. Thus, this work proposes representing an image as a sequence of object-level information, extracting highly semantic features from models pre-trained on an object-centric dataset, in order to feed a bidirectional Long Short-Term Memory network trained for scene classification. We perform a Many-to-Many training approach, such that each input outputs a corresponding scene prediction, allowing us to use each individual prediction to boost recognition with a weighted voting approach. To the best of our knowledge, our sequence representation, as well as our late fusion of predictions, was little pursued by methods from the literature based on recurrent approaches for scene recognition. We evaluated our proposal on three widely known datasets for scene recognition: Scene15, MIT67 and SUN397, outperforming recurrent-based methods on MIT67, a dataset entirely dedicated to the problem of indoor scenes, while the others, which mix indoor and outdoor environments presented as a greater challenge for our approach. However, we were able to improve performance on all datasets over the most successful methods on the literature by pairing our work to a few of them in an ensemble of classifiers. Meaning a joint strategy with our method was beneficial for the task of scene classification.

Keywords: Computer Science, Artificial Intelligence, Image Recognition, Neural Networks

List of Figures

1.1	Comparison between indoor and outdoor environments to illustrate the value of object composition for indoor scenes.	2
1.2	Diverse applications for scene recognition.	4
2.1	Illustration of a forward pass for a recurrent unit.	7
2.2	Different training procedures supported by Recurrent Neural Network. . .	8
2.3	Representing the flow of information through an LSTM unit.	9
3.1	Scene recognition methods distinguished by low-level and semantic modeling as proposed by Bosch et al. [2007]. Image extracted from Bosch et al. [2007].	13
3.2	Dendrogram of distances between models as proposed by Fei-Fei and Perona [2005].	14
3.3	Multi-scale Orderless Pooling (MOP-CNN) proposed by Gong et al. [2014]. Image extracted from Gong et al. [2014].	15
3.4	Results from Zhou et al. [2014a] for scene recognition pre-trained on object-centric and scene-centric data.	15
3.5	Quad-directional sliding window approach from the work of Zuo et al. [2015] to model spatial context in images.	18
3.6	Hierarchical recurrent approach proposed by Zuo et al. [2016].	18
3.7	Methodology proposed by Javed and Nelakanti [2017]. A combination of CNN and RNN architectures to model spatial context for scene recognition.	19
4.1	Overview of our methodology.	22
4.2	Comparing number of regions proposed by Selective Search for classes with different amounts of objects.	24
4.3	Expanded representation of a BiLSTM.	27
4.4	Expected appearance of our weight matrix proposition.	28
4.5	Overview of ensemble approach.	30
4.6	Composing vector p^{max} of maximum activations from all timesteps.	31
5.1	Samples from each class of Scene15. Image from Lazebnik et al. [2006]. . .	36
5.2	Samples from MIT67 divided into 5 main categories for better visualization of the broad variety of scene classes [Quattoni and Torralba, 2009].	36
5.3	Samples from SUN397, divided into 3 major categories: indoor, urban and nature [Xiao et al., 2010].	37

5.4	Average number of patches proposed by Selective Search before and after filtering for Scene15	38
5.5	Average number of patches proposed by Selective Search before and after filtering for MIT67	39
5.6	Example of coverage percentage analysis for a single image.	39
5.7	Average coverage percentage of patches from MIT67 after filtering.	40
5.8	Average coverage percentage of patches from Scene15 after filtering.	41
5.9	Bar plot representing the amount of non-zero cells on each row of our weight matrix for Scene15.	42
5.10	Top-5 object weights for all classes of Scene15.	43
5.11	Weight vectors from matrix W^{obj} at the columns corresponding to three different object categories.	43
5.12	Bar plot representing the amount of non-zero cells on each row of our weight matrix for MIT67.	44
5.13	Top 30 object weights for bathroom, bedroom, living-room and kitchen on MIT67.	45
5.14	Feature importance results for each empirically selected statistical measure. Results for all three datasets.	46
5.15	Random decision tree from the output Random Forest.	47
5.16	Percentage predicted by our method and by paired classifiers according to the switch criteria.	48
5.17	Variations of recurrent approaches for scene recognition in order to analyze the contribution of each aspect of our method.	51
5.18	Per-class performance on MIT67 with a vanilla majority voting compared with our weighted approach.	54

List of Tables

5.1	Results with vanilla majority voting compared to our weighted approach. .	45
5.2	Accuracy results for the ensemble of paired classifiers.	48
5.3	Comparing the accuracy our proposed approach with methods from the literature.	50
5.4	Comparing different training approaches and recurrent architectures for scene recognition on Scene15.	52
5.5	Comparing different training approaches and recurrent architectures for scene recognition on MIT67.	52

List of Acronyms

CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
GRU	Gated Recurrent Unit
ROI	Region of Interest
M2M	Many-to-Many
M2O	Many-to-One
BiLSTM	Bidirectional Long Short-Term Memory
SVM	Support Vector Machine

Contents

Acknowledgments	vii
Resumo	ix
Abstract	xi
List of Figures	xii
List of Tables	xiv
List of Acronyms	xv
1 Introduction	1
1.1 Motivation	3
1.2 Problem Definition	5
1.3 Document Structure	6
2 Theoretical Foundations	7
3 Related Work	11
3.1 Scene Recognition	11
3.1.1 Recurrent Neural Network	17
4 Methodology	21
4.1 Composing a Sequence of Object Parts	23
4.1.1 Feature Extraction	25
4.2 Context Modeling with a BiLSTM	25
4.3 Weighted Majority Voting	27
4.4 Ensemble of Classifiers	29
5 Experiments and Results	35
5.1 Datasets	35
5.2 Scenes as Sequences of Objects	37
5.3 LSTM Settings	40
5.4 Object Weights	41
5.5 Ensemble of Classifiers	45

5.6	State-of-the-Art Scene Recognition	49
5.7	Ablation Analysis	51
6	Conclusions	55
6.1	Future Work	56
	Bibliography	57

Chapter 1

Introduction

The ability to recognize the environment around us might seem effortless for humans, but research on scene recognition shows otherwise for computers. According to Xiao et al. [2010], a scene is defined as any place a human being can act within or to which one could navigate, ranging from house rooms to islands, stadiums, cathedrals, among many others. Scene recognition is still regarded as an open challenge, even though there are methods that surpass human-level accuracy [Wang et al., 2017]. Different from other classification tasks, such as recognizing objects or faces, scenes can be quite hard. Besides the usual challenges such as lighting, angle of image acquisition, occlusion, to name a few, the image of a scene can be abundant in highly variable local information. According to Quattoni and Torralba [2009], this variability is specially true for indoor scenes.

While vanilla classification methods show good performance on scenes from an outdoor environment, the performance decreases for indoor categories. Quattoni and Torralba [2009] attribute this behaviour to the fact that indoor scenes are well represented by the objects they contain. Each object can present itself in a variety of manners, and their disposition in the environment can also be very diverse, putting indoor scene recognition as an even harder challenge, requiring approaches specially tailored for the task. For a more intuitive understanding on the differences between indoor and outdoor environments, refer to Figure 1.1. It is noticeable how outdoor scenes are usually composed of large structures while small components on indoor places convey valuable information when seen together. The distinction between indoor and outdoor environments has been known in the literature for a long time, with early methods of scene recognition successfully attempting to classify images between those two categories [Szummer and Picard, 1998; Serrano et al., 2004]. But only a decade ago the field of indoor scene recognition started receiving dedicated attention.

With the rise of Convolutional Neural Network (CNN) [LeCun et al., 1999] as the most promising approach for classification on images, many attempts have been made to tackle the issue of scene recognition for both indoor and outdoor environments with similar techniques. With the introduction of a large scale scene-centric dataset [Zhou et al., 2014b] the expectations were even higher for a CNN to be the best solution. However, even though the results were promising, future works gained greater prominence by taking advantage of high level semantic knowledge, usually conveying



Figure 1.1: Comparison between indoor and outdoor environments to illustrate the value of object composition for indoor scenes. On the top we see three indoor classes: bedroom, office, and bar. On the bottom the categories are: mountain, industry, and beer garden.

object-level information and their intricate relationship [Wang et al., 2017; Herranz et al., 2016; Nascimento et al., 2017].

More recently we witnessed the surge of Recurrent Neural Network (RNN) and its variations [Sherstinsky, 2018]. The ability to correlate information from parts of a sequence was designed to solve a whole new class of problems, mainly the ones that presented temporal dependencies. Text [Pérez-Ortiz et al., 2001], audio [Wöllmer et al., 2010], and time sequences such as stock market prices [Hsieh et al., 2011] were the primary types of data in which an RNN was applied. And, as expected, they benefited a lot from the behaviour of that type of model. However, any data that can be divided into interdependent parts is eligible to exploit the advantages of recurrent models.

To think of a single image as sequential data requires us to determine which level of representation is relevant to the goal. For instance, semantic segmentation methods based on recurrent approaches use a single pixel as the atomic part of the sequence, modeling correlation between pixels in order to predict the label of a single pixel based on its surrounding context [Shuai et al., 2016; Byeon et al., 2015]. As previously mentioned, the literature presents successful approaches on scene recognition by correlating object information. Hence, this Thesis proposes to classify scenes using a methodology based on a type of RNN, a Long Short-Term Memory (LSTM) [Hochreiter and Schmidhuber, 1997]. Specifically, we exploit the advantages of a Bidirectional Long Short-Term Memory (BiLSTM), which provides predictions of higher quality compared to its unidirectional counterpart [Schuster and Paliwal, 1997]. We work with

the assumption that scenes can be well represented by their composition of objects; therefore, in this Thesis we consider object-level information to compose the sequence.

The main idea is to use a region proposal method to output the Region of Interest (ROI) in the image, extract high level features from each ROI and feed it to the recurrent model. Since the order of elements is a main factor for recurrent models, it needs to be consistent throughout samples, thus we chose a region proposal method, called Selective Search [Uijlings et al., 2013], which sorts the proposed regions by the likelihood of it containing an object. The use of automatic region proposal with significant order of parts along with extracting high quality object-level features allow us to work with a recurrent model without any object label or bounding boxes from the input scene. By training an LSTM to perform scene classification on the proposed object parts, our goal is to model context from objects optimizing it to provide semantic meaning to the correlated parts. We also specify a Many-to-Many (M2M) training procedure, i.e., producing output scene predictions for each object part, allowing us to boost recognition performance through a weighted majority voting that takes into account the relevance of each part to the evaluated scene. Finally, this Thesis also shows that by paring our proposal with approaches from the literature in an ensemble of scene classifiers it is possible to outperform existing methods, leading to the conclusion that a joint strategy with our method is beneficial even for the best performing approaches of scene recognition reported on the literature.

1.1 Motivation

Scene recognition is a research field with an extensive amount of applications. For instance, it is usually related to the field of robotics as a fundamental perception task, as mentioned in the work of Liao et al. [2016]. Mobile robot autonomy is an important goal in this field, and a better semantic understanding of the environment through vision can educate the robot as to how it should behave in each environment, being it through navigation, object manipulation, or high level tasks such as intelligent conversations with humans in the environment. For instance, the work of Espinace et al. [2013], depicted at Figure 1.2, proposes to exploit the advantages of an indoor mobile robot, such as its additional sensors, to provide a more accurate classification of the environment. It is based on the assumption that objects correlate to scene categories, which is one of the basis of our work. We can expand this application definition as developing a system that will interact with the real world. Knowing the environment can level the expectations of possible occurrences and adequate behavior.

Many works also mention the benefits of knowing scene categories to facilitate object classification, since scene-object relationship is very discriminative for both tasks. This reciprocity is found in works that use object labels and probabilities to recognize scenes [Liao et al., 2016; Espinace et al., 2013; Li et al., 2010] or the other way around, exploiting scene knowledge to predict object classes [Grzeszick and Fink, 2016]. The latter is specially true for multi-label classification tasks, when the goal is to classify multiple objects on the same image [Li et al., 2018].

Another application worth mentioning is image retrieval, a field of great importance to encourage methods of scene categorization, which reduced the search space for

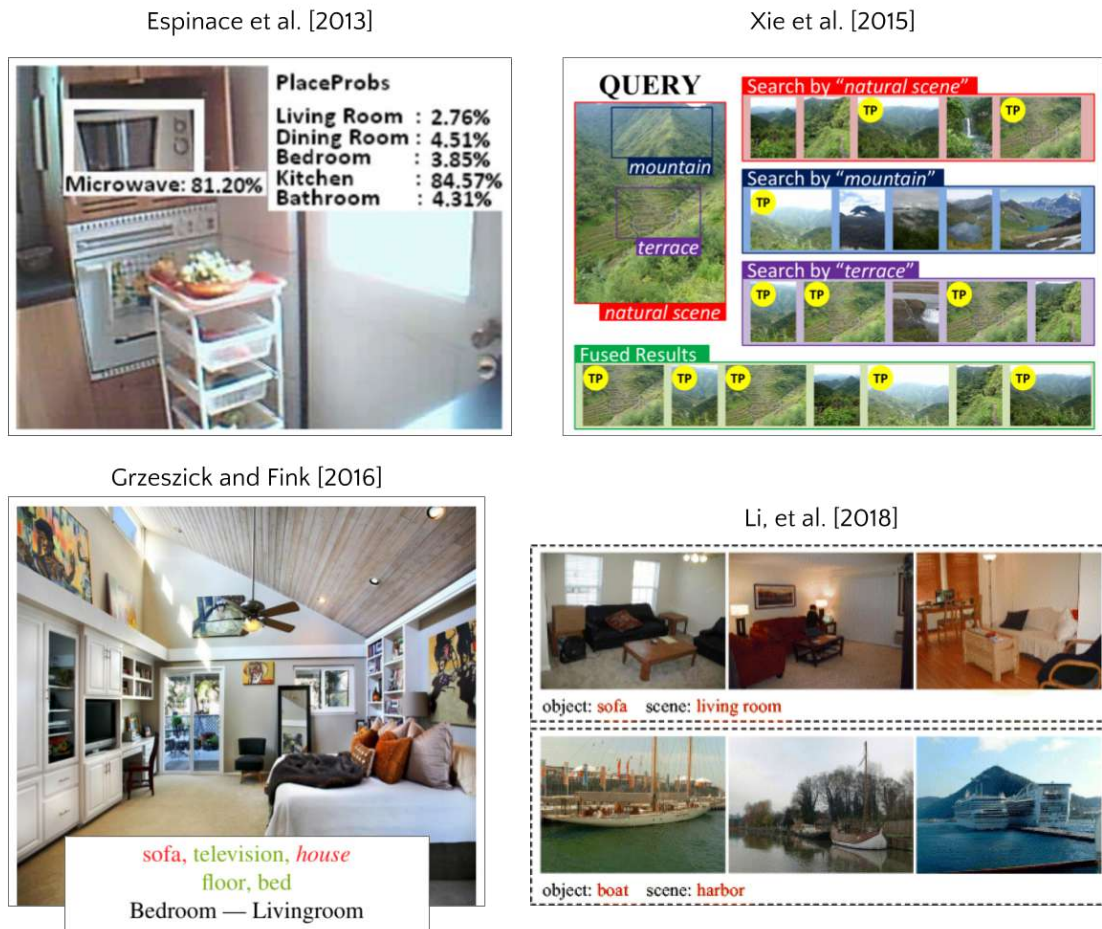


Figure 1.2: Diverse applications for scene recognition. On the top left we have the work of Espinace et al. [2013] with mobile robot navigation, followed by Xie et al. [2015], top right, modeling image retrieval as problems of scene and object recognition. Bottom left shows Grzeszick and Fink [2016], a zero-shot object recognition using scene knowledge, and finally Li et al. [2018] exploiting scene cues to improve multi-label classification. Images extracted from [Espinace et al., 2013; Xie et al., 2015; Grzeszick and Fink, 2016; Li et al., 2018].

a given query to the related category of images. Figure 1.2 shows the work of Xie et al. [2015]. The authors refer to both problems of image retrieval and scene recognition as one, meaning they can be tackled with similar strategies. The aforementioned applications compose only a small set of possibilities that scene recognition allows, hence it is one of the greatest research challenges known to the fields of Computer Vision, Machine Learning, and the ones related to them.

Apart from applications, another motivation for this work is the overwhelming amount of visual content available online. It is one of the main reasons that caused a growth in research on scene recognition, as well as related topics on that field of research. Scene understanding soon became a very active topic and started to quickly evolve. The volume of easily acquired images allows for research on many visual methods, specially data hungry ones like most deep learning approaches.

Finally, it was only recently that different types of RNN were applied to problems of image recognition, thus little is known about the advantages and disadvantages of using such models. Further research is required in order to understand the kind of information recurrent models convey for images and the potential for understanding the image as a composition of parts on a semantic level.

1.2 Problem Definition

As previously mentioned, there are evidences that indoor scenes are well represented by the composition of object-level information. The intuition behind methodologies which follow that premise is that object parts do not independently allow the inference of scene categories, on the other hand, their correlation is discriminative enough to distinguish between classes. That assumption permits us to think of a scene as a collection of object-level information which can be modeled as sequence data, if we assume a meaningful order for the parts. RNNs are built for sequential input, given its quality to correlate sequence elements, modeling its intricate structure. For scene images, the underlying structure of object parts can be interpreted as contextual information.

The literature still has room to research different methodologies that can exploit the advantages of a recurrent approach for problems of image classification, particularly for indoor scene recognition. One of our goals is to contribute for that branch of literature. Thus, we are interested in modeling the problem of scene recognition as a correlation of object-level information using an RNN-based methodology. Specifically, we defined three different questions this work aims to tackle:

- How to provide a high quality representation of a scene as a sequence of interdependent object parts without object labels?
- Which recurrent configuration and training procedure are pertinent to the problem of indoor scene recognition?
- Can the information from each individual object part boost recognition?

Furthermore, we experimented with an ensemble of paired classifiers as a joint strategy of our proposal and some of the most successful approaches on the literature, in order to evaluate if our method can improve over each of them. We evaluate classification performance on three datasets, Scene15 [Fei-Fei and Perona, 2005], MIT67 [Quattoni and Torralba, 2009] and SUN397 [Xiao et al., 2010], each presenting a different level of difficulty. And finally, as an additional goal, we expect to encourage further research on the use of recurrent models applied to scene recognition, given their potential to encode contextual information.

As an additional contribution, results from the present research were published at the 30th Conference on Graphics, Patterns and Image, SIBGRAPI [Laranjeira and Nascimento, 2017].

Thesis Statement:

Recurrent Neural Network are beneficial when tackling the problem of indoor

scene recognition, since such scenes are characterized by interdependent object-level information. Additionally, a bidirectional approach can provide an output response for each object part relative to the remaining context of the image, permitting a boost on recognition performance.

1.3 Document Structure

The remaining chapters are organized as follows. Chapter 2 outlines the theoretical foundations required for a better understanding of our proposition. Following, Chapter 3 contains a literature revision on scene recognition as a whole and the use of recurrent models. Next, Chapter 4 describes the methodology for scene recognition with Recurrent Neural Networks. Chapter 5 refers to the experimental setup and the discussion of our findings. And finally Chapter 6 concludes and establishes future steps.

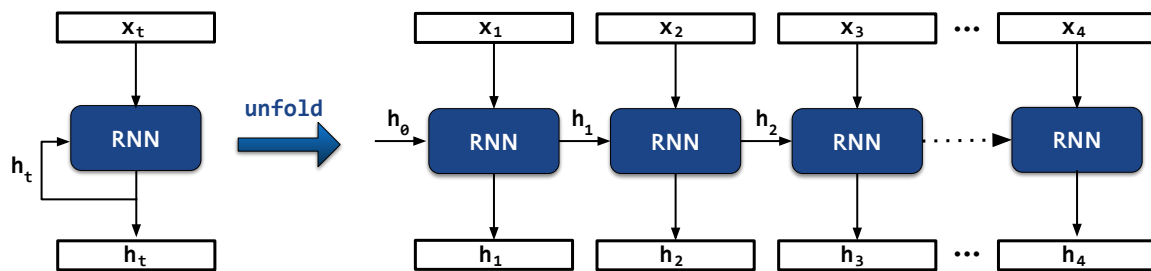


Figure 2.1: Illustration of a forward pass for a recurrent unit. Each timestep t receives an element \mathbf{x}_t , part of the sequential input, generating a hidden state \mathbf{h}_t , which serves both as an output and an additional input for the next timestep $t + 1$.

Chapter 2

Theoretical Foundations

For a better understanding of our proposition, this chapter provides the reader with foundations of Recurrent Neural Network. It also contributes to a more clear understanding of how such models can contribute to the literature of indoor scene recognition.

RNNs were designed to model structure from a sequential input as a solution for problems with temporal dependencies. Figure 2.1 depicts the behaviour of a simple recurrent unit. The input is a sequence $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ of features $\mathbf{x}_t \in \mathbb{R}^d$, where d is the number of dimensions from each feature. The inputs are fed iteratively to the RNN unit. At every iteration t , which is called a timestep, the recurrent unit receives the corresponding input \mathbf{x}_t according to the input order of elements, and outputs an \mathbf{h}_t . The latter is its internal memory, which is passed back to the RNN through a feedback loop, feeding the next timestep. Specifically, a recurrent unit is a function of the current input \mathbf{x}_t and its past memory \mathbf{h}_{t-1} , accumulating knowledge as it receives new information so that the inference at a timestep t takes into account all past information. Figure 2.1 depicts this behaviour presenting two different views, to the left there is the loop representation, since there is only a single recurrent unit that receives one input at a time, updating its internal state. To the right the same unit is unfolded for n timesteps, showing a more intuitive representation of the iterations on a recurrent unit.

From Figure 2.1 one can infer that for each input, an RNN generates a correspond-

ing output. In practice, there are a few ways of modeling a sequence-based problem, all showed in Figure 2.2. From left to right, the first one is a regular feed forward network, that receives one input and generates a single output, hence the name One-to-One. It does not actually encodes structural information, since there is no sequence involved. Secondly, the One-to-Many training procedure receives only a single input and generates a sequential output. For instance, the problem of image captioning feeds the RNN with an image as a single input, training it to generate a sequence of words describing the image. It is important to realize that even though there is no input from the second timestep on, the output of each timestep is a feedback to the RNN itself, serving as input to generate further outputs. The Many-to-One is the most common format of a recurrent modeling, since it is suitable for sequence classification, such as Sentiment Analysis, or image classification, as our main problem states. It receives sequential input, and it is trained to generate a single label for the entire sequence. Following, the Many-to-Many training procedure, sequence input and sequence output, can be presented in two different formats: regular M2M and synchronized M2M, differing only by the input-output correspondence. While a regular M2M does not necessarily associate each output to an input, e.g., Text Translation, the synchronized counterpart is defined by such correspondence, where every \mathbf{x}_t has an \mathbf{h}_t directly related to it.

Given its characteristics, the RNN does not require a fixed number of elements on the input sequence. The model is capable of accumulating knowledge by correlating any given number of inputs, as well as generating any number of outputs, with no mandatory constraints. Its main limitation is regarding the vanishing/exploding gradient problem, a very common issue for recurrent approaches, since they can unfold into a large amount of timesteps depending on the sequence length. During backpropagation, long sequences may cause the gradients to either explode or tend to zero, thus not accumulating knowledge from earlier timesteps [Hochreiter, 1998]. That problem motivated researchers to propose improved recurrent units capable of encoding long

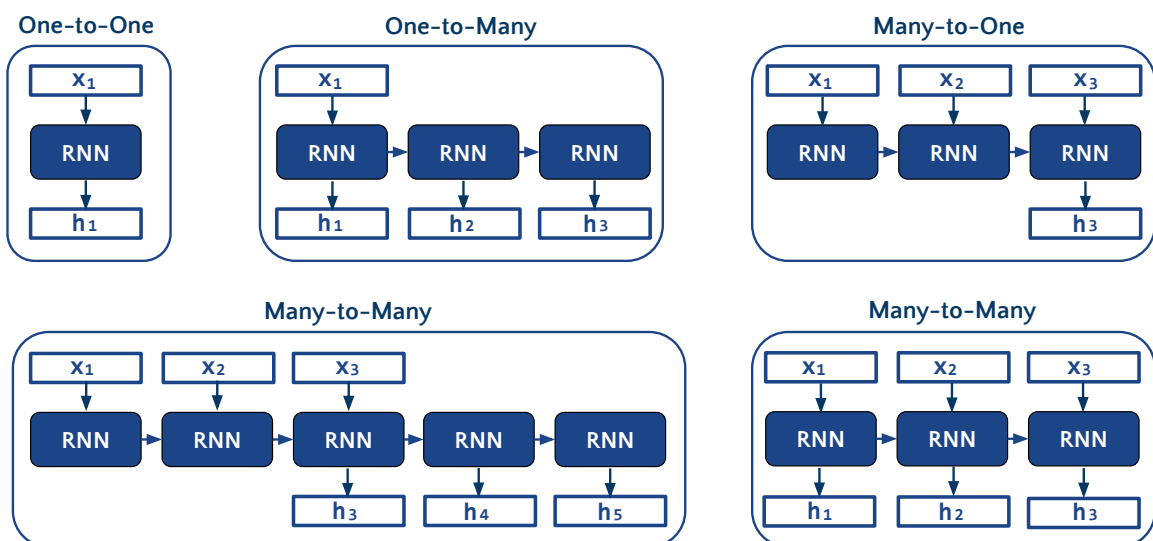


Figure 2.2: Different training procedures supported by Recurrent Neural Network.

term dependencies, such as Gated Recurrent Unit [Cho et al., 2014] or Long Short-Term Memory unit [Hochreiter and Schmidhuber, 1997]. This is accomplished by a more constant update of their internal memory through minor linear operations, such that its gradient is less likely to vanish or explode.

In order to understand the methodology proposed by this thesis, it is necessary to expand the concept of recurrent units. Specifically, let us focus on LSTM units. Figure 2.3 illustrates the flow of data through cells. An LSTM is a recurrent unit that produces an output hidden state (\mathbf{h}_t) as any other recurrent unit, but its unique aspect is the cell state C_t , an internal memory calculated through minor linear operations, making it stable throughout iterations, therefore providing the LSTM with the ability to retain long-term information, avoiding the vanishing/exploding gradient issue [Hochreiter et al., 2001].

Therefore given an input sequence $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$, in order to produce \mathbf{h}_t and C_t for each timestep t , the LSTM relies on three gates: forget gate f_t , input gate i_t and output gate o_t , which are basically structures responsible for deciding how much of the information will follow through and how much is blocked. The construction of each gate, presented in Equation 2.1, is mainly a linear operation comprised of two weight matrices W^x and W^h that multiplies the current input \mathbf{x}_t and the previous hidden state \mathbf{h}_{t-1} respectively, and also a bias b . Additionally, each gate has a nonlinear activation function to allow non-linearities in the sequence to be modeled. In addition to the three gates, Equation 2.1 also shows the calculation of candidate cell states \tilde{C}_t , which is also comprised of two weight matrices and a bias, and will later generate the final internal memory of the unit. Those four structures comprise all the trainable information of an LSTM unit, amounting to a total of eight weight matrices and four biases to be optimized during training. The calculation of gates and candidate cell states is defined

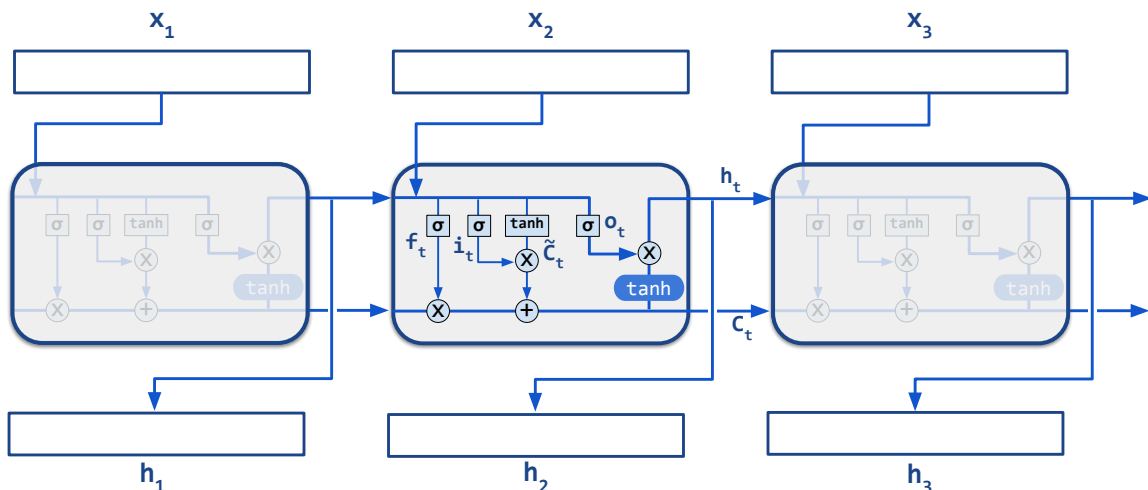


Figure 2.3: Representing the flow of information through the LSTM. Each gate is built to filter information before it is passed forward. The stable update of its cell state C_t avoids a vanishing/exploding gradient.

as follows:

$$\begin{aligned}
 f_t &= \sigma(W_f^x \mathbf{x}_t + W_f^h \mathbf{h}_{t-1} + b_f), \\
 i_t &= \sigma(W_i^x \mathbf{x}_t + W_i^h \mathbf{h}_{t-1} + b_i), \\
 \tilde{C}_t &= \tanh(W_C^x \mathbf{x}_t + W_C^h \mathbf{h}_{t-1} + b_C), \\
 o_t &= \sigma(W_o^x \mathbf{x}_t + W_o^h \mathbf{h}_{t-1} + b_o).
 \end{aligned} \tag{2.1}$$

After building the gates and candidate cell states, the recurrent pass of timestep t is concluded by applying each gate to filter a different source of information. Forget gate f_t (refer to Figure 2.3) is responsible for filtering previous cell states C_{t-1} , which is commonly interpreted as "forgetting the past". Input gate i_t filters the candidate cell states \tilde{C}_t calculated on the current timestep, representing which new information will be incorporated into the output cell state C_t . Equation 2.2 shows how C_t is calculated with f_t and i_t :

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{2.2}$$

where \odot is the element-wise product of vectors.

Finally, hidden state \mathbf{h}_t is the main output feature produced by the LSTM. In other words, any layer after that on a deeper architecture would receive some or a combination of all \mathbf{h}_t as its input. Its calculation is mainly based on applying the output gate (o_t) to filter information from the internal cell state C_t , as follows

$$\mathbf{h}_t = o_t \odot \tanh(C_t). \tag{2.3}$$

Chapter 3

Related Work

In this thesis, we investigate the use of a recurrent approach to perform scene recognition. Hence, this chapter focuses on two main topics from the literature. The first section outlines the advances throughout the years for scene classification, from early methods mainly composed of handcrafted low-level features such as color and texture, to the more recent debut of deep learning approaches, which mostly comprises CNN-based methods, but recently led to the use of recurrent architectures. Followed by Section 3.1.1 detailing the main literature on applying Recurrent Neural Networks to image related problems, for instance scene labeling by modeling correlation among pixels, and the plethora of combinations of Convolutional and Recurrent models applied to image classification.

3.1 Scene Recognition

Scene Recognition has been an active field for over a couple of decades. Over the late 90's and early 2000, the field of image retrieval was of great importance to newly proposed scene recognition approaches. For instance, Vailaya et al. [1998] propose an automatic categorization of a given image between two categories: city and landscapes, on the interest of improving image retrieval approaches. The goal was to decrease the search space over the database of images, comparing the query only to images from the corresponding category. The authors produced a dataset entirely dedicated to outdoor images, and relied on low-level features describing color, texture and edge information. Vailaya later published another study derived from the first one, proposing a hierarchical classification approach also relying on low-level features, in order to provide more specific labels to further improve the approaches for image retrieval. The hierarchy followed from indoor/outdoor classification, to classifying outdoor images as city or landscape, and finally classifying landscape images into more specific categories: sunset, forest and mountains [Vailaya et al., 2001]. For each level of hierarchy, different low-level features were extracted from the images, feeding a pre-trained Bayes decision tree in order to perform classification.

Furthermore, the work of Ulrich and Nourbakhsh [2000] was applied to robot localization. The problem was also modeled based on image retrieval references. The

authors constructed a database of reference images from the desired environments, and during operation its place recognition module compared the acquired image, i.e., the query, to references on the database. The matching was performed with a nearest neighbor approach based on color histograms for each individual band on the chosen color spaces.

Human unconscious behavior described through the view of Psychology was also very useful on the early developments of scene recognition, as showed by the work of Oliva and Torralba [2001]. They propose what is called a GIST descriptor, a concept borrowed from psychology, which says scenes can be quickly recognized by its 'gist', i.e., its essentials such as spatial layout of unspecified elements, or even volumetric forms, with no need to represent the specifics of the environment. The descriptor was referred by the authors as a spatial envelope, encoding semantic characteristics based on human perception: roughness, naturalness, openness, etc. It neglects object information as a requirement for scene recognition, encoding only global aspects of scenes, which was found by the authors to perform poorly on indoor environments.

Bosch et al. [2007] noticed that approaches from the literature roughly followed one of two methods for scene representation, the first one based on modeling low level features from the entire image or from sub-blocks, as aforementioned with the works of Vailaya et al. [1998], Ulrich and Nourbakhsh [2000] and Vailaya et al. [2001]. And a second one relying on semantic information from the image, being it from objects or more abstract concepts, such as those presented by Oliva and Torralba [2001]. Figure 3.1 shows an illustration of the proposed distinction. Early works already realized the benefits of adding semantic cues to improve classification over methods solely based on low-level features Serrano et al. [2004]. With time, intermediate semantic representations became more common on the literature, since they performed better than its counterpart.

Mid-level representations were also widely exploited by researchers in the context of scene recognition. For instance Fei-Fei and Perona [2005] proposed a Bag-of-Words representation, a concept widely used with text data [Blei et al., 2003], which had been previously adapted to the field of Computer Vision [Csurka et al., 2004]. Fei-Fei and Perona [2005] built a codebook of visual features from the training set by clustering patches from multiple scales with a K-means algorithm [Lloyd, 1982], and classifying new images according to the activation pattern among existing codewords. It was an entirely unsupervised proposition, showing competitive performance against heavily annotated supervised approaches. However, the authors noticed their approach lack information required by indoor environments, since the four indoor categories showed the highest error rate, as illustrated in Figure 3.2, a dendrogram of pseudo-euclidean distance between models, in which all four indoor classes are closer together, which led the classification to higher confusion among them. The authors attributed the poor behavior on indoor scenes, to man-made environments sharing a few characteristics encoded by their approach, such as sharp horizontal and vertical edges.

Many other works brought up the concept of Bag-of-Words applied to images, entitled Bag-of-Features [Sivic and Zisserman, 2003]. One of the highlights from the literature is the work of Lazebnik et al. [2006], a Spatial Pyramid Matching approach that produced multiple Bag-of-Features from three scales. The idea was to provide

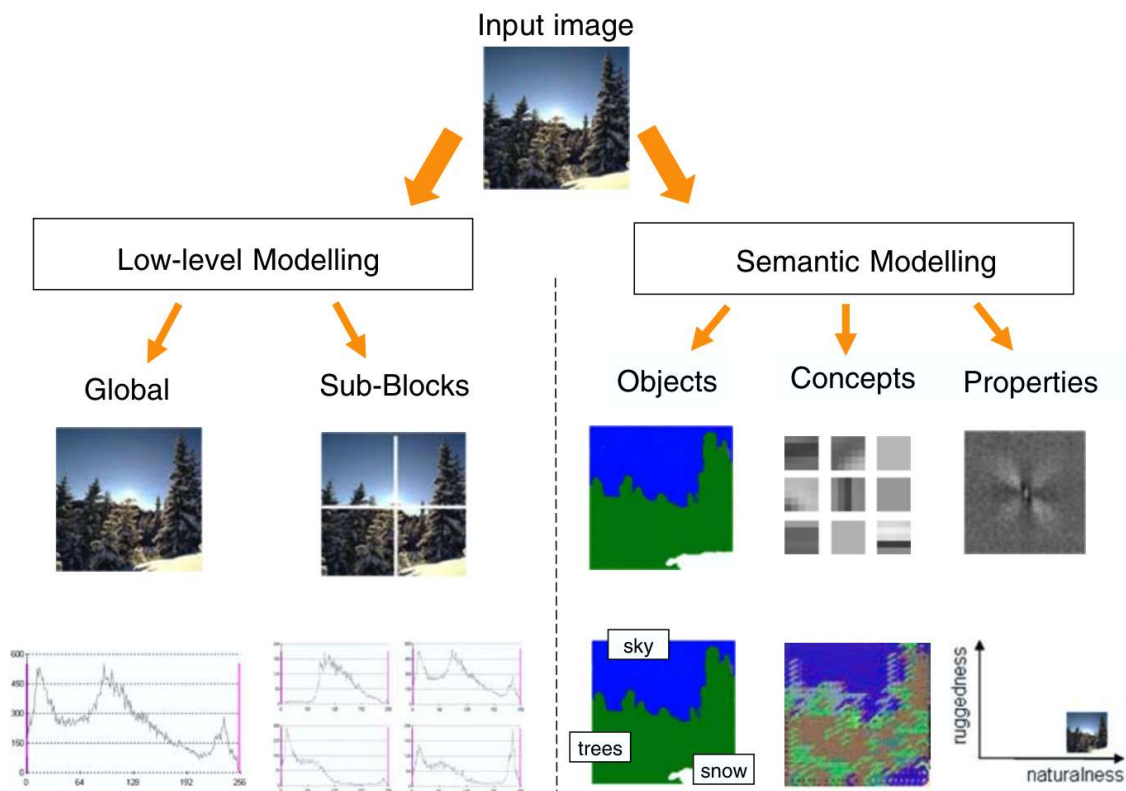


Figure 3.1: Scene recognition methods distinguished by low-level and semantic modelling as proposed by Bosch et al. [2007]. Image extracted from Bosch et al. [2007].

global scene cues from higher scales to inform the search for specific objects on local regions. They managed to outperform previous scene recognition methods while also providing good quality object recognition. On the other hand, they also presented the same weakness as previous methods: indoor scenes. Although they did not discuss this specific problem, indoor classes showed a performance much lower than average.

The distinction between indoor and outdoor scenes was already perceived by early methods from the literature, dedicated to the problem of classifying between both categories [Ssummer and Picard, 1998; Serrano et al., 2004], even realizing the greater challenge when tackling indoor environments [Oliva and Torralba, 2001]. Although more sophisticated approaches were proposed, the problem of indoor scenes was still present, since most approaches relied mainly on global aspects of the image to perform classification. But only years later the field of indoor scenes would rise as an individual research problem, for which datasets and approaches were entirely dedicated to it [Quattoni and Torralba, 2009].

The popularization of deep Convolutional Neural Network raised the bar on average performance for scene recognition. Early CNN approaches were directed to the problem of object recognition, specially after the release of a large-scale object-centric dataset, ImageNet [Deng et al., 2009]. Researchers adapted existing object recognition methods, making them suitable for other tasks, in order to take advantage of the high quality features provided by a pre-trained deep model. For scene recognition, in

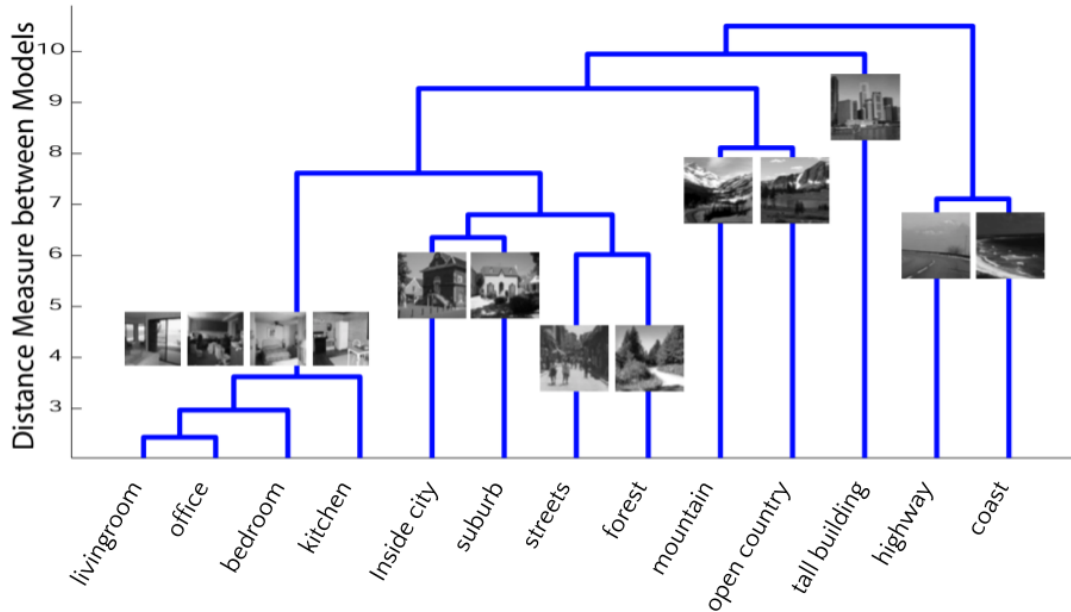


Figure 3.2: Dendrogram of distances between models as proposed by Fei-Fei and Perona [2005]. Indoor scenes are closer together causing greater confusion to the classification approach. Image extracted from Fei-Fei and Perona [2005].

the work of Gong et al. [2014], depicted by Figure 3.3, a multi-scale pooling approach was proposed, extracting features from three different levels of the input image using the convolutional architecture from Jia et al. [2014], and concatenating the pooled results for each scale. The convolutional model was pre-trained on ImageNet, achieving 51.98% on SUN397 and 68.88% on MIT67, showing a performance largely superior to most methods on the literature. The authors highlighted their results on MIT67 as very relevant, since they focus on representing a combination of global and local information, which is suitable for the problem of indoor scene recognition. Sharif Razavian et al. [2014] achieved a similar result on MIT67 simply by training an Support Vector Machine (SVM) classifier with features extracted from a single scale (the entire image) using the publicly available CNN entitled OverFeat [Sermanet et al., 2013], also pre-trained on ImageNet. They also highlighted that such features can be applied as an off-the-shelf approach for several applications other than scene recognition.

After a large-scale scene centric dataset was released, entitled Places [Zhou et al., 2014a], there was a lot of investment in CNN approaches as the solution to the problem of scene recognition. Models pre-trained on Places showed great improvement over the state of the art, reaching over 79% on MIT67, almost 92% on Scene15 and over 63% on SUN397, all with a VGG architecture [Simonyan and Zisserman, 2014], as showed in Figure 3.4. And although the debut of Places was groundbreaking, classifying scenes was still regarded as an open challenge, and researchers were still providing solutions tailored to the specific task of recognizing indoor environments.

As proposed by Gong et al. [2014], combining local information of a given scene was shown very promising in the literature, specially for indoor scenes. A similar

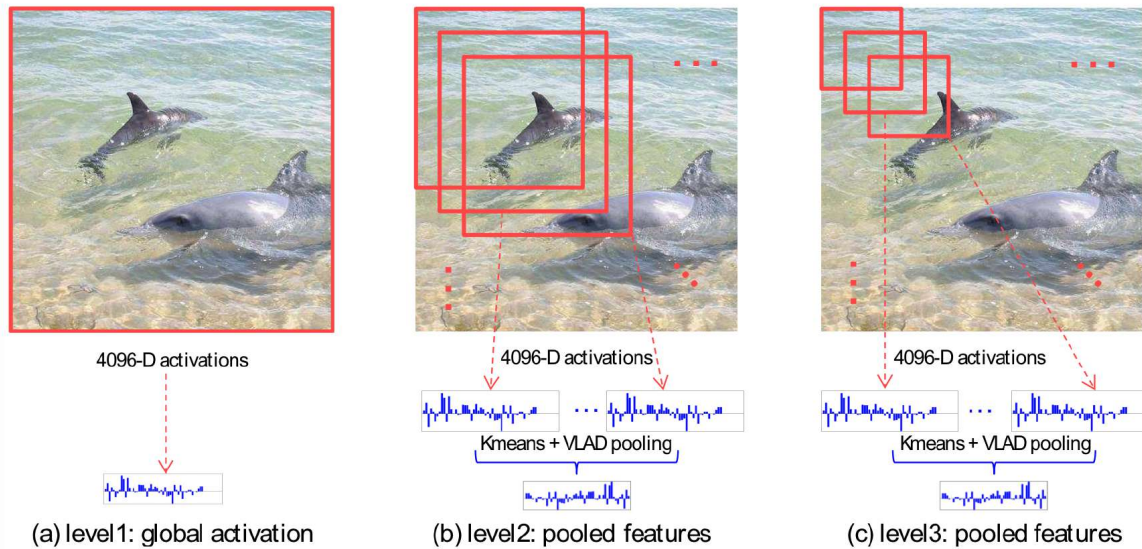


Figure 3.3: Multi-scale Orderless Pooling (MOP-CNN) proposed by Gong et al. [2014]. Image extracted from Gong et al. [2014].

Deep Feature	SUN397	MIT Indoor67	Scene15	SUN Attribute
Places365-AlexNet	56.12	70.72	89.25	92.98
Places205-AlexNet	54.32	68.24	89.87	92.71
ImageNet-AlexNet	42.61	56.79	84.05	91.27
Places365-GoogLeNet	58.37	73.30	91.25	92.64
Places205-GoogLeNet	57.00	75.14	90.92	92.09
ImageNet-GoogLeNet	43.88	59.48	84.95	90.70
Places365-VGG	63.24	76.53	91.97	92.99
Places205-VGG	61.99	79.76	91.61	92.07
ImageNet-VGG	48.29	64.87	86.28	91.78
Hybrid1365-VGG	61.77	79.49	92.15	92.93

Figure 3.4: Results from Zhou et al. [2014a] for scene recognition with the most successful CNN architectures at the time, pre-trained on object-centric and scene-centric data. Models pre-trained on Places outperform previous methods with a classic CNN + SVM approach. Table extracted from Zhou et al. [2014a].

approach was later proposed by Herranz et al. [2016], only this time with a joint strategy of object features for local scales and scene-level features for the entire image. Herranz et al. [2016] explored several combinations of models pre-trained on ImageNet and others trained on Places, to validate its intuitive premise that object-level information was mostly relevant to describe local information. Herranz et al. [2016] outperformed the proposition of Zhou et al. [2014a], also relying on a VGG architecture as feature extractor, isolating the impact of their proposition over a method based only on scene-level information. Nascimento et al. [2017] followed the same premise on combining scene-level and object-level information on different scales. They build a dictionary of

deep object features on multiple scales followed by a sparse coding approach activating sparsely over the dictionary. The authors highlight the robustness to occlusion and artificial noise of their approach, mostly attributing it to the sparse feature composition. Finally, there is also the work of Wang et al. [2017], proposing an architecture entitled PatchNet, adapted from VGG and Inception V2 [Normalization, 2015], which provides patch-level appearance through the extraction of highly semantic features from the last convolutional layer of their architecture, and the aggregated object probabilities from its prediction layer. Both outputs serve as input for their newly proposed encoding approach, entitled Vector of Semantically Aggregating Descriptor (VSAD). They share the premise of composing a representation with rich local information, such that indoor scenes will not represent a weakness of their approaches.

This section did not include information on RNN-based approaches, although they started to be explored as an alternative to model correlation of local information over the past few years. Section 3.1.1 will outline and properly contextualize recurrent approaches within the timeline of scene recognition development.

3.1.1 Recurrent Neural Network

When Recurrent Neural Network started gaining popularity for image-related problems, it was common to see them applied to inherently sequential data, such as handwriting recognition [Liwicki et al., 2007]. It is reasonable since they were designed for such problems, but soon enough researchers started rethinking challenges such as image classification, segmentation or even synthesis, making them suitable for a recurrent approach. For images, modeling the structure of parts is equivalent to learning contextual dependencies, which is highly valuable for problems requiring correlation of local information.

As letters are the atomic unity of a text, pixels are the equivalent for images. Thus, applying RNNs to correlate pixel-level information lead to significant advancements on several research fields such as scene labeling [Byeon et al., 2015; Shuai et al., 2016] and image completion [Oord et al., 2016]. The work of Oord et al. [2016] is an important reference for generative models since it was capable of predicting missing pixels from images, to a level of filling half occluded samples with high quality results. As for scene labeling, correlating pixels with a recurrent approach allow for a fine-grained segmentation of semantic parts of the image with models of much lower computational complexity.

Along with pixel-level correlation, there are also significant references working on higher levels of images, more suitable to problems such as scene recognition. One reference that stands out on the literature of scene recognition is the work of Zuo et al. [2015], one of the first reports applying a combination of Convolutional and Recurrent layers to correlate semantic features from several regions of the input image. Their method, entitled C-RNN, was competitive to the state of the art at the time, achieving 68.50% on MIT67 and 51.14% on SUN397. In order to model a single image as a sequence of parts, a quad-directional sliding window approach was adopted, as illustrated by Figure 3.5. Since they worked with intermediate features of size 6×6 from a Convolutional architecture, the recurrent approach consisted of iterating through each row (or column, depending on the direction), generating six outputs for each direction, exactly as represented by the Figure. The fully connected layers that complemente their architecture is responsible for generating class probabilities. Their entire network was trained on object-centric data from ImageNet and fine-tuned for each scene dataset.

Extending the work of Zuo et al. [2015], in 2016 the authors attempted a hierarchical approach [Zuo et al., 2016], entitled C-HRNN, following similar steps to their previous attempt. The method was also based on intermediate Convolutional features, a recurrent modeling of such features and fully connect layers generating class probabilities. Only this time the RNN in the middle operated over multiple scales, encoding spatial dependencies intra-scale and transferring information onto the corresponding regions of higher scales. They also improved over their previous quad-directional flow

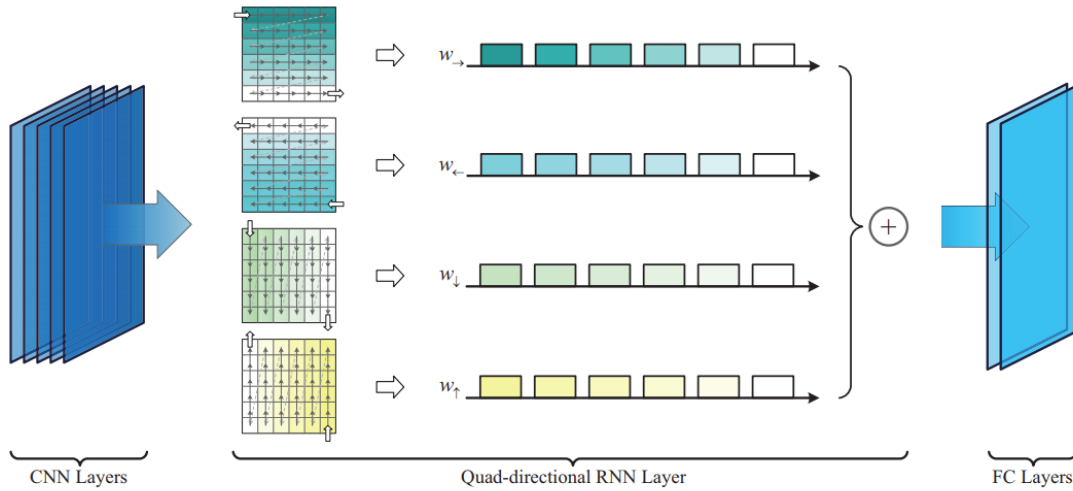


Figure 3.5: Quad-directional sliding window approach from the work of Zuo et al. [2015] to model spatial context in images. Image extracted from Zuo et al. [2015].

of information, maintaining the sliding window approach, but eliminating contextual abrupt interruptions such as skipping from the last sliding window on the right of a row into the left-most window of the next row. Despite the significant effort, their proposal pre-trained on ImageNet showed only a slight improvement over Zuo et al. [2015], gaining 1.6 percentage points for SUN397 and less than 1 percentage point for MIT67. However, by pre-training on a scene-centric dataset, Places, 60.34% and 75.67% on SUN397 and MIT67 respectively.

It is also worth mentioning the work of Javed and Nelakanti [2017], a recent proposal for scene recognition that also assumes object features as an ideal source of information to recognize scenes, reinforcing our premise. This work showed that with a small fraction of a large-scale dataset, one can be competitive with the state of the

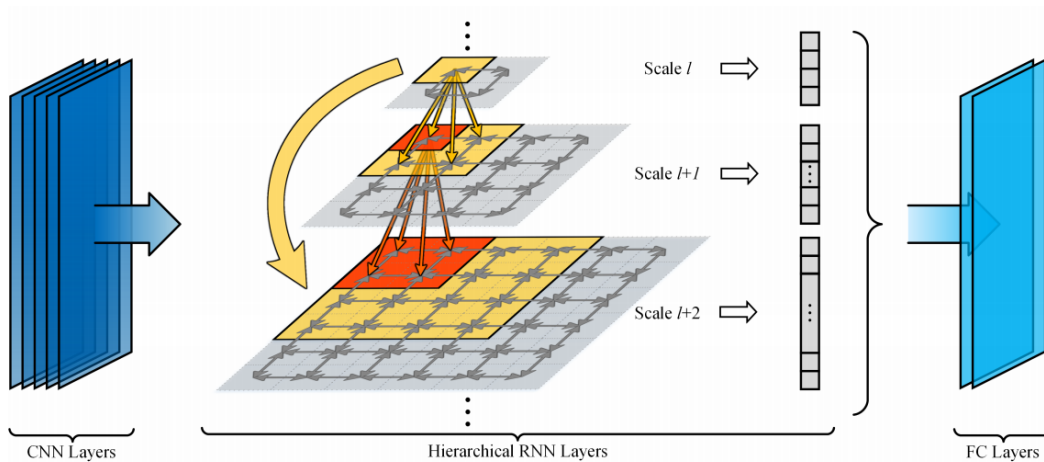


Figure 3.6: Hierarchical recurrent approach proposed by Zuo et al. [2016]. Image extracted from Zuo et al. [2016].

art by correlating the objects of a scene. As input for the RNN, this approach suggests selecting Region of Interest from the image with a region proposal algorithm, ordering the bounding boxes by the algorithm's confidence score, decreasingly. As illustrated by Figure 3.7, the recurrent step is performed on the last convolutional layer of a CNN architecture at the respective location of the original bounding boxes. On their experiments the number of ROI was fixed to 10, arguing that it was sufficient as a proof of concept to validate the methodology. Since recurrent models allow inputs of variable size, fixing the number of ROI omits an important aspect of the scene. The amount of object information present in each scene by itself conveys relevant knowledge regarding its category, i.e., some classes can be typically more crowded than others. Additionally, by fixing the number of bounding boxes, the approach risks leaving behind important information, since the confidence score of a region proposal algorithm is not optimized for the problem of scene recognition, thus it does not take into account how relevant the objects are for each scene category.

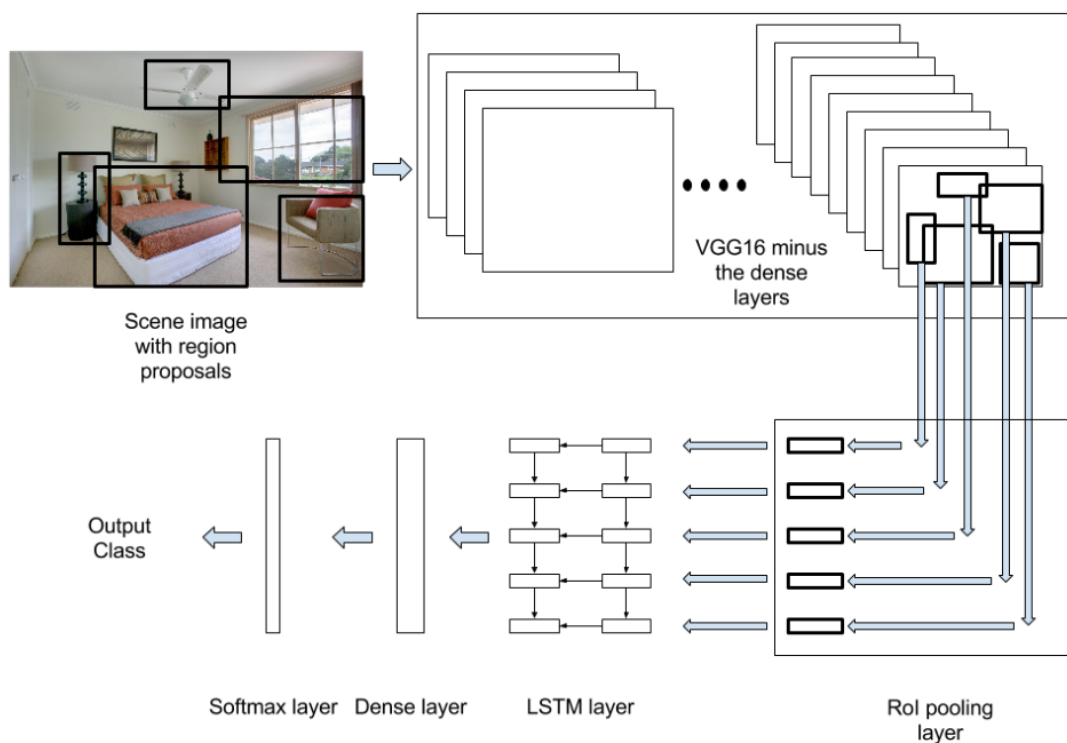


Figure 3.7: Methodology proposed by Javed and Nelakanti [2017]. A combination of CNN and RNN architectures to model spatial context for scene recognition. Image extracted from Javed and Nelakanti [2017].

Chapter 4

Methodology

This chapter describes the proposed methodology, as illustrated by Figure 4.1. In order to build an approach for scene recognition based on a recurrent model, we need to represent an image as a sequence of elements. Thus, step (a) (refer to Figure 4.1) of the methodology is dedicated to dividing the image of a scene into parts of scene objects, ordered by significant criteria in the interest of composing a sequence. Then, since our premise is based on representing an image by its composition of objects, for each part we extract high-level object features from a deep CNN, constituting step (b) of our method, composing a sequence $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ of features $\mathbf{x}_t \in \mathbb{R}^d$ where d is the dimension of our chosen deep feature, as it will be detailed later. The composed sequence serves as input to our recurrent model, step (c). We propose an Many-to-Many training approach for a Bidirectional Long Short-Term Memory such that each sequence element \mathbf{x}_t produces an output y_t based on the current input along with accumulated context of the remaining parts. Since we only have scene-level labels, all outputs are an attempt at predicting the category y of an input scene.

At test time we add steps (d) and (e). The first one generates a single prediction y'_{our} from the recurrent model through a weighted majority voting. We expect to boost classification performance relative to a vanilla voting approach, since not every part of the scene is equally relevant. The voting weights are based on pre calculated object weights representing the relevance of each object class for a given scene category, as will be described later on this chapter.

Finally, step (e) is an ensemble with our method and a paired classifier. Here, the certainty of our prediction is measured, allowing us to eventually resort to the output y'_{paired} from the paired classifier whenever our confidence is too low. Thus, the final output y' is the result of a switch criteria based on statistical measures over our own predictions, allowing it to be paired with any scene classification approach.

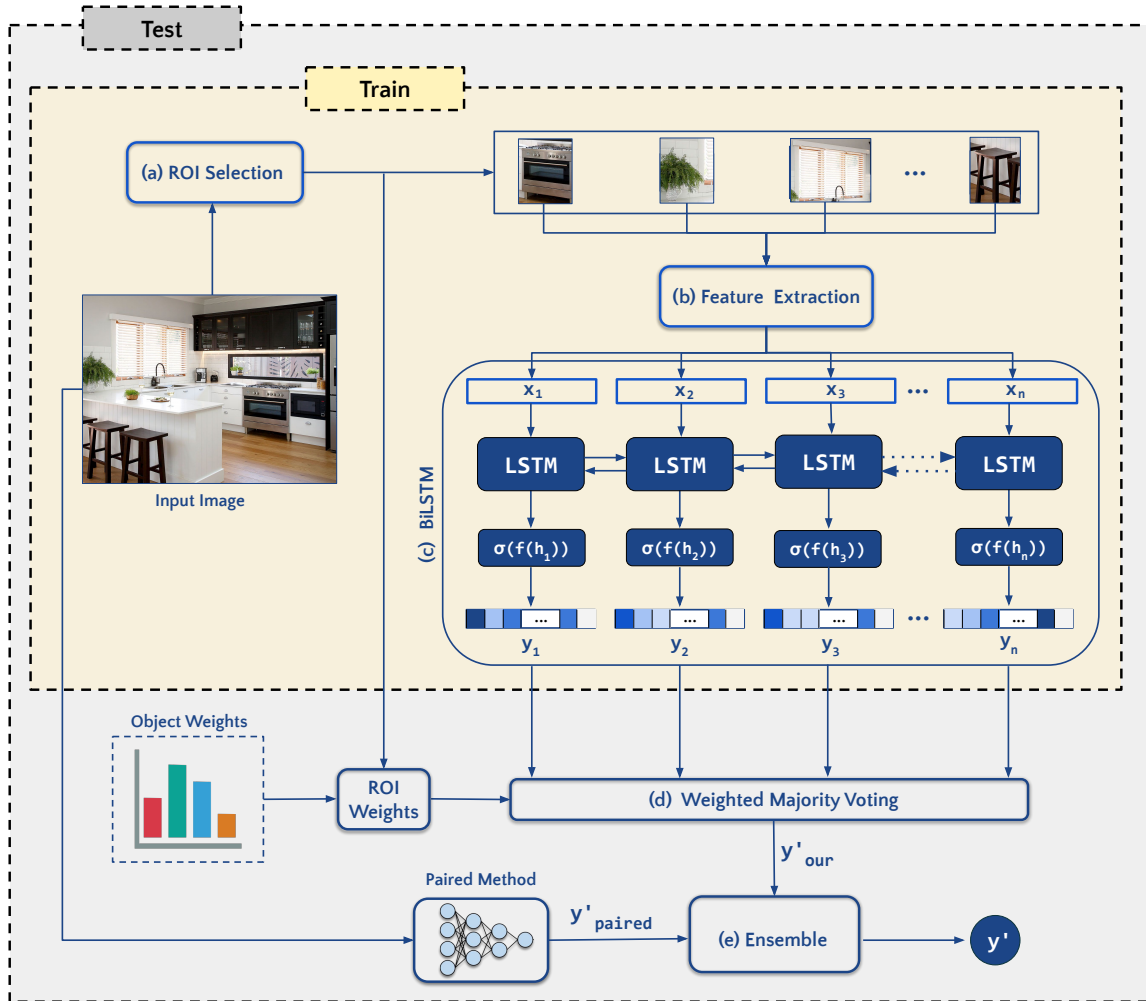


Figure 4.1: Overview of our methodology. Steps (a) through (c) constitute the training steps, respectively (a) dividing the image into object parts; (b) extracting high level features from each part; and (c) training a Many-to-Many BiLSTM to produce a prediction y_t for each x_t . At test time, step (d) performs a majority voting weighting each prediction by the semantic relevance of the corresponding patch, outputting a prediction y'_{our} . Finally the ensemble of classifiers, step (e) decides between our prediction and a paired classifier from the literature.

In summary, our approach is divided into five steps:

- (a) Composing a sequence of objects or object parts;
- (b) Extracting high-level object features;
- (c) Training a M2M BiLSTM;
- (d) Weighted voting of predictions;
- (e) Ensemble with paired classifier.

The remaining of this chapter is organized in four sections. The first one details steps (a) and (b), composing the sequence that serves as input for the recurrent model. Section 4.2 describes the context modeling with our proposed recurrent approach, representing step (c). Section 4.3 outlines the steps to perform a weighted majority voting, and finally Section 4.4 outlines in details the ensemble of classifiers.

4.1 Composing a Sequence of Object Parts

The goal of our first step is to compose an ordered sequence of ROI from the image, containing interdependent object parts. In Chapter 3, we discussed a few ways the literature has attempted to represent a single image as a sequence, and it is important to highlight that for a recurrent model this a very important aspect since it is built to model correlation among the input parts. Considering that we do not have available annotations on object labels and bounding boxes for scene images, we chose a well-known algorithm for object proposals called Selective Search [Uijlings et al., 2013], which yields 99% recall, meaning it selects nearly all object information from the scene. It computes a hierarchical segmentation, grouping adjacent segments by similarity, iteratively, adding to the list of proposed regions at every computation step, i.e., it outputs regions on different scales of the image.

Seeing that the Selective Search algorithm outputs object bounding boxes, it is intuitive to infer that depending on the characteristics of the scene, the number of output regions can vary drastically. Figure 4.2 illustrates that behaviour by showing the number of regions selected for scenes with different amounts of object-level information. For classes such as *deli*, scenes are usually crowded with delicacies up for sale, while categories like *pool inside* present fewer objects other than the pool itself. This is relevant because it means the output sequence based on a region proposal approach has variable length. To the best of our knowledge, there are only a couple of works on the topic of scene recognition which exploits such an approach to represent a single image as a sequence, and they choose to fix the sequence length despite the aforementioned behavior of region proposal methods [Javed and Nelakanti, 2017; Wang and Pan, 2017].

It should also be noted that the number of ROI proposed by Selective Search can reach hundreds or even thousands of bounding boxes, which is roughly presented in Figure 4.2 and will be further explored at Chapter 5. And since scene recognition has large datasets, by using all the proposed ROI we would be handling a massive set of patches, which would be intensely time consuming for a proper training. Therefore, to compose a smaller and more feasible sequence we filter the proposed bounding boxes by their size relative to the entire image. The idea is to define two thresholds t_{lower} and t_{upper} representing the lower and upper percentage limits of patch size. Selective Search provides the size of each segment in pixels, which we will call s_{patch} , as an attribute of the output. Thus given the image size s_{img} as the product of its width and height, we allow patches within the following range:

$$s_{img} * t_{lower} < s_{patch} < s_{img} * t_{upper}. \quad (4.1)$$

The output of Selective Search is decreasingly ordered by the likelihood of a region



Figure 4.2: Comparing number of regions proposed by Selective Search for classes with different amounts of objects. The categories presented are deli (top) and pool (bottom) from MIT67, respectively containing 1, 259 and 385 proposed patches.

to actually contain an object, which we will be calling *objectness*. We maintain the algorithm's order of elements when composing our sequence, meeting the requirement of a consistent order of elements throughout all samples. The final output of this step is a sequence of bounding boxes from the filtered output of Selective Search, decreasingly ordered by objectness.

4.1.1 Feature Extraction

After selecting all ROI from the image, the next step is the extraction of highly semantic features from each region. Since our main goal is to input a recurrent model with a sequence of object-level information from the image, the process of feature extraction should convey information of that nature. Deep learning approaches are powerful feature extractors for many applications, and this is specially true for object features. Residual nets were able to take the representation performance even further by adding residual functions to allow training of deeper networks [He et al., 2016]. This outstanding performance decreased the error rate on the ImageNet challenge (ILSVRC) [Deng et al., 2009] to 3.75%. We exploit the advantages of its 50-layer variation, entitled Resnet-50, to serve as feature extractor of our methodology. We perform a forward pass on Resnet-50 pretrained on ImageNet, and extract the last convolutional layer, after average pooling, providing a highly semantic and discriminative object feature of $d = 2,048$ dimensions. After extracting features from each region, the final output of that step is a sequence of object features $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ with $\mathbf{x}_t \in \mathbb{R}^d$, ordered according to the objectness criteria, defined on the previous step. Refer to Figure 4.1 for a visual representation of the sequence composition.

It is worth mentioning that our choice of input representation can be replaced, for instance using different feature extraction methods, or even different criteria to divide the image into interdependent parts. The imperative factor of the recurrent input that define our work and commit to our premise is composing a sequence of object-level information from the image, allowing the recurrent approach to model contextual information by correlating object parts.

4.2 Context Modeling with a BiLSTM

Once the input scene is represented as a sequence \mathcal{X} of features, our goal is to model the image context by correlating all $\mathbf{x}_t \in \mathcal{X}$. We propose to exploit the power of recurrent models to represent the structure of the sequence. Therefore, step (c), presented in Figure 4.1, consists in training a variation of a Recurrent Neural Network optimizing it for classification, since the model will learn the structure of scenes, producing similar intermediate representations for samples from the same category, i.e., the modeled structure will convey semantically meaningful information of the scene.

On the choice of an RNN variation, there are already studies that evaluate the performance of different recurrent units. For instance, Chung et al. [2014] highlights that gated units are in fact superior to a vanilla unit. Mainly, gated units allow long-term context modeling due to their ability to avoid the vanishing/exploding gradient problem. As for the difference between the two advanced gated units, i.e., Gated Recurrent Unit (GRU) [Cho et al., 2014] and LSTM [Hochreiter and Schmidhuber, 1997], no significant performance gap was found. Thus, we chose the LSTM variation due to the more extensive literature successfully applying it to different kinds of data.

It is important to notice that an LSTM approach, as any other recurrent approach, can be deep in time, or more generally in sequence length. However, the number of parameters is limited to the weights of a single recurrent unit, since it it-

erates through all timesteps using the same weights. Its size depends solely on the input size and a hyperparameter that determines the size of the hidden state. Hence, a recurrent approach produces models with smaller sizes compared for example to a deep CNN.

More generally, an LSTM unit is a function of the current input and previous knowledge. Hence it is capable of remembering past information and accumulate knowledge throughout iterations. Although it was designed for data with inherent sequential structure, when applied to images it correlates the given parts just as it would for any other data. As long as the input has structured dependencies between parts, a recurrent approach is capable of modeling it.

Let us elucidate a little further the characteristics of our sequential input built by steps (a) and (b). Each element conveys object-level information and the sequence as a whole has a consistent order in every sample, the objectness. However there is no compulsory direction the recurrent model should follow. There is no beginning and ending defined by our data. Since the unidirectional LSTM accumulates knowledge at every iteration from in a given direction, it begins with no information whatsoever, hence one could argue that it should be fed first with the most representative data as an early boost to acquire relevant context. Although the opposite argument could also be defended, there is a better solution for such situation. We can exploit the advantages of a bidirectional approach [Schuster and Paliwal, 1997], which can accumulate knowledge from both directions and produce a better informed inference.

As defined by Schuster and Paliwal [1997], a bidirectional approach is based on training simultaneously on positive and negative directions, and it is suitable whenever the entire sequence is available at once during inference, producing superior results compared to the unidirectional variation. In practice, a bidirectional recurrent approach means having two recurrent units, each one accumulating knowledge from a different direction. As a result, at every timestep t there is information available from the entire image, the sequence "past" (positive direction) and the "future" (negative direction), which means an output produced at iteration t is a function of the current input and the context of the remaining sequence elements, parts of an image in our case. Based on that, we use a synchronized Many-to-Many (M2M) training procedure, producing a scene classification output y_t for every input \mathbf{x}_t . Since each element of our sequence has meaningful semantic information from objects, each prediction can potentially convey information of how such element relates to its context.

Figure 4.3 illustrates the behavior of the bidirectional LSTM used in this work. Each unit receives data at opposite orders and calculates Equations 2.1 to 2.3, accumulating knowledge from different directions. As a consequence, every timestep t has two hidden states, one for the positive direction \mathbf{h}_t^+ and one for the negative \mathbf{h}_t^- . In order to perform a single inference, the final hidden state \mathbf{h}_t is produced by Equation 4.2, defined as

$$\mathbf{h}_t = \mathbf{h}_t^+ \oplus \mathbf{h}_t^- \quad (4.2)$$

with \oplus representing a concatenation. A fully connected layer receives \mathbf{h}_t as input, followed by a softmax activation to produce the probability vector of categories. In such case, given the hidden size of each recurrent unit, the subsequent dense layer

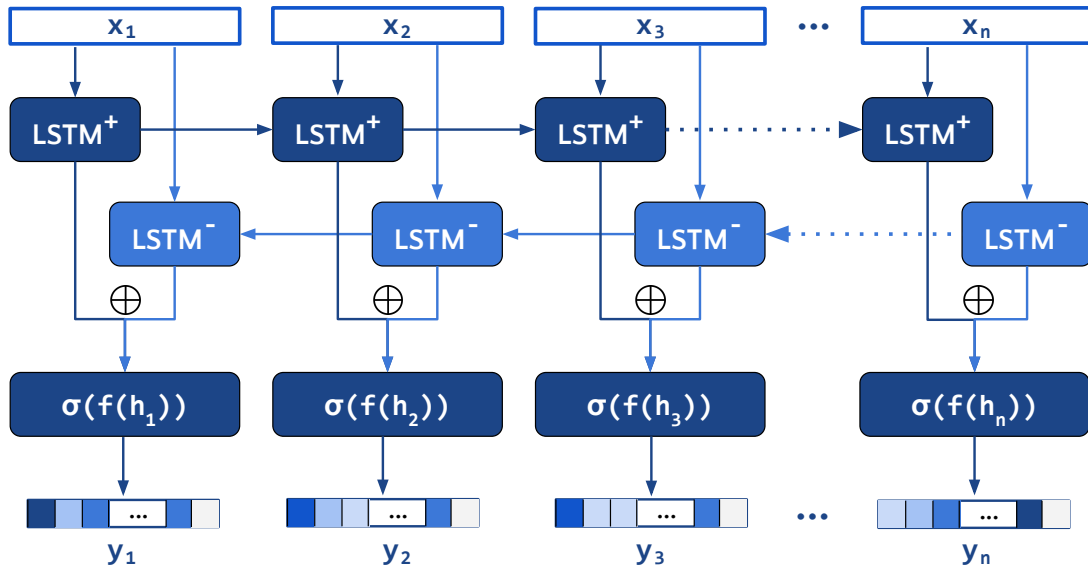


Figure 4.3: Expanded representation of a BiLSTM. The output of each timestep is a concatenation of the output from both LSTM units.

must have twice the number of parameters to support bidirectional output from the BiLSTM, i.e., $2 \times \mathbf{h}_t$.

As a synchronized M2M procedure, every \mathbf{h}_t is forwarded through the fully connected layer and activated with a softmax in order to produce one prediction for each input. Likewise, the loss calculation should take into account errors from all timesteps. Since we are optimizing our model to perform classification and we only have scene-level labels, our loss for each timestep t is a Cross-Entropy function between the probability vector y_t and a one-hot encoding representing the scene category y , according to Equation 4.3,

$$\ell(y_t, y) = - \sum_i (y_t^i \log(y^i) + (1 - y_t^i) \log(1 - y^i)). \quad (4.3)$$

At a high level of abstraction, considering that each input \mathbf{x}_t is directly related to a patch from the scene, a prediction at time t represents how patch t relate to the remaining context, and consequently how it affects prediction. The final loss is then calculated as an average of every $\mathcal{L}(y_t, y)$ calculated previously, as Equation 4.4 shows:

$$\mathcal{L}(y', y) = \frac{1}{n} \sum_{t=1}^n \ell(y_t, y). \quad (4.4)$$

4.3 Weighted Majority Voting

Seeing that after trained our methodology generates n predictions, n being the number of ROI selected from a scene, we still need to output a single prediction to perform in-

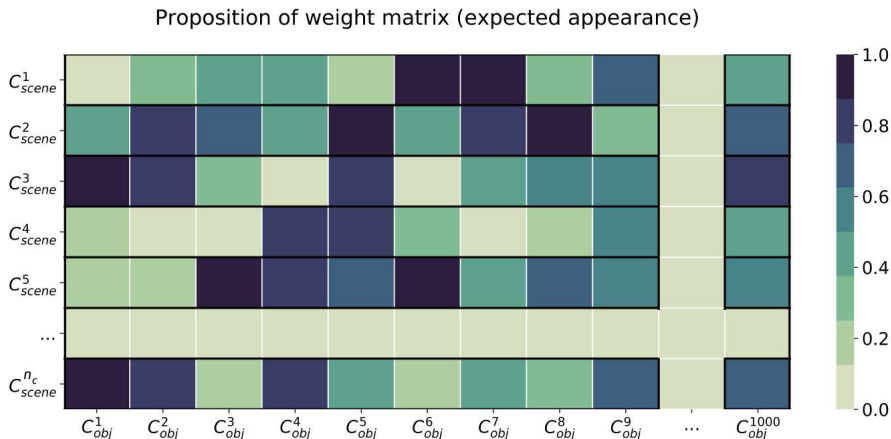


Figure 4.4: Expected appearance of our weight matrix proposition. Each cell (i, j) corresponds to the relevance of object j to class i .

ference on test samples. To do so, we aggregate all inferences throughout iterations as a weighted majority voting. The rationale is that not all regions are equally important to discriminate a category of scene, and a weighted voting can improve the classification performance. And although the LSTM itself should learn the proper contribution of each patch, we want to add explicit and interpretable information regarding the relationship between scenes and objects. Thus, our proposal is that the weights reinforce such information, which is analogous to the priori of a Bayesian inference model, while the predictions serve as evidence that will provide a posteriori.

To calculate the weights, we use a validation set to build a weight matrix W^{obj} of size $n_c \times n_o$, respectively the number of scene classes on the dataset and the number of all possible objects, for which we considered all $n_o = 1,000$ categories from Imagenet. An illustration of the expected appearance of matrix W^{obj} is presented in Figure 4.4. The rows are represented as C_{scene}^i for a scene category i , while the columns C_{obj}^j correspond to each object j . Consider the pair (i, j) a cell on row i and column j of our matrix.

We start by initializing W^{obj} with all zeros. Then, we gradually fill it such that for every ROI feature \mathbf{x}_t from the input image, first we predict to which object class j it belongs, and then we find out the relevance of j for the scene class i , given by the scene label. Since our features \mathbf{x}_t are from the last convolutional layer of a Resnet-50 pre-trained on ImageNet, to acquire an object prediction j we forward our feature through the prediction layer of Resnet-50, which outputs a probability vector y_t^{obj} of object classes. The maximum activation from y_t^{obj} corresponds to the predicted object class j for a given \mathbf{x}_t . Since we do not have object labels on any of the datasets, we rely on the prediction power of Resnet-50 to provide potential labels.

Knowing to which object class a given feature \mathbf{x}_t potentially belongs, in order to represent the relevance of object j to the scene category i , we forward \mathbf{x}_t on our already trained recurrent model, generating a probability vector y_t of class predictions. It is worth recalling that our model was trained with a M2M approach, hence the fusion of prediction is only performed at test time, when it is required to generate a single

inference y'_{our} . Given that we are aware of the scene’s true label i when composing W^{obj} , we increment its cell (i, j) by the probability of class i according to y_t , defined by y_t^i , i.e.,

$$W_{i,j}^{obj} = W_{i,j}^{obj} + y_{t,i}. \quad (4.5)$$

The rationale is that y_t^i corresponds to the probability of object j belonging to class i . Of course, a BiLSTM takes into account the entire context to output a prediction, but one of our assumptions is that a M2M recurrent model allows to isolate the exerted influence of a part relative to the whole.

Once W^{obj} is entirely filled by all samples from our set, as a normalization approach we divide the weights from each cell by the number of patches from the correspondent class used to fill each row of the matrix. This will benefit objects that occur more often, which is also an important aspect on the relevance of such object. It is noteworthy that such a matrix has to be constructed for each individual dataset, in order to encode any particularities it may have.

At test time, we perform a weighted majority voting between predictions from all timesteps, using matrix W^{obj} to provide the weights. Given an input $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ of features, from each \mathbf{x}_t we predict the object class j , and the corresponding recurrent prediction y_t of size n_c . Let w_t^j represent the j^{th} column of W^{obj} at timestep t . Our weighted prediction will then be defined by Equation 4.6,

$$\hat{y}_t = y_t \odot w_t^j, \quad (4.6)$$

where \odot represents the element-wise product of both vectors. Afterwards, the strongest activation from each \hat{y}_t contributes as the vote for class i at iteration t , such that

$$v_{t,i} = \begin{cases} 1, & \text{if } \hat{y}_t \text{ voted for class } i \\ 0, & \text{otherwise.} \end{cases} \quad (4.7)$$

We then aggregate the votes for each class, given by $v_i = \sum_{t=1}^n v_{t,i}$ with i varying from 1 to the number of classes n_c , and n representing the number of patches from the given scene. The final prediction y'_{our} is given by class i with a larger voting sum.

4.4 Ensemble of Classifiers

The prediction y'_{our} from the previous step is sufficient to perform scene recognition, however, we are also interested in knowing if our method adds any information over the state of the art. For that purpose, we propose to pair our own approach with methods from the literature, based on a switch criteria that will determine for a given image which of the paired approaches should be considered the final output prediction. We trained a Decision Tree [Breiman et al., 1984] with statistical measures over our predictions, as it will later be outlined in details. If the measures indicate a weak prediction, the paired approach will provide a prediction. Since we chose state-of-the-art approaches as paired classifiers, our goal is to see if our reliable predictions can improve over a few of the best approaches in literature. Figure 4.5 is an overview of

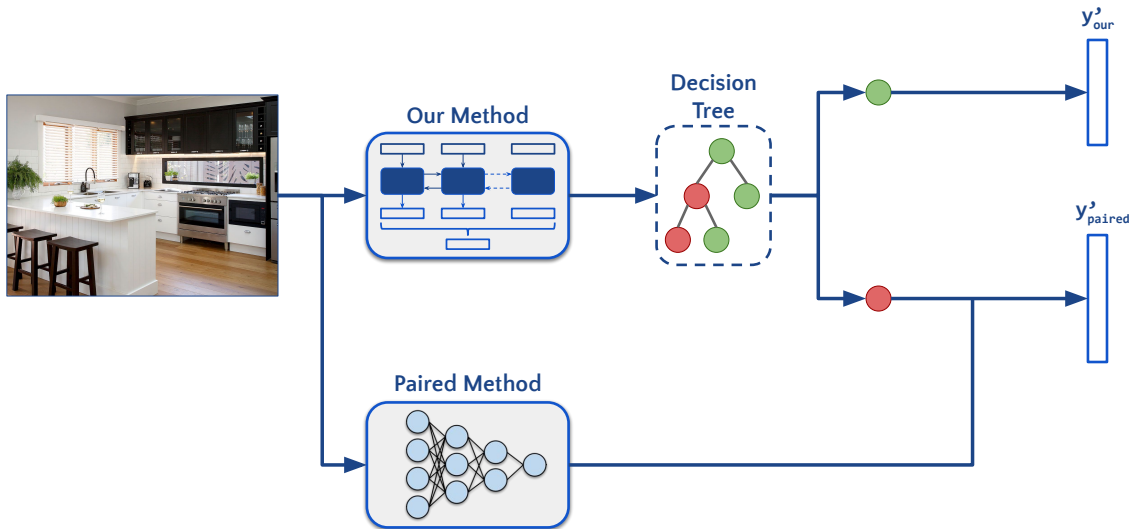


Figure 4.5: Overview of ensemble approach. The decision tree is a switch criteria to determine prediction reliability. Green circles represent a reliable inference while red circles indicate the paired approach should provide the prediction. [Better seen in colors]

our proposal. It is important to notice that we only apply the switch criteria over our own method, hence our ensemble can be paired with any classifier regardless of their particularities.

The main portion of this step is to determine a switch criteria capable of measuring the reliability of our prediction. Since we are working with a recurrent approach, the distribution of predictions throughout timesteps shows great potential to convey such information. Thus, we propose to construct a unidimensional vector p^{max} by extracting the maximum activation of the probability vectors from all timesteps, i.e., each p_t from $p^{max} = \{p_1^{max}, p_2^{max}, \dots, p_n^{max}\}$ corresponds to $\max y_t$ from timestep t . That is equivalent to a unidimensional max pooling, a sample-based discretization approach that outputs the maximum value of the given input, reducing its dimensionality. In our case, the rationale is that a maximum activation from a given y_t is valuable information regarding the prediction distribution on such timestep, since the components of y_t always add up to 1. For instance, a high $\max y_t$ indicate a higher level of certainty on the output prediction, while lower $\max y_t$ means the prediction was a little more fuzzy on timestep t . The final output is a vector of size n (number of patches from the scene), as illustrated by Figure 4.6.

Once p^{max} is calculated, we extract a few statistical measures empirically chosen in order to train our decision trees to discriminate accurate predictions from misclassifications on a validation set. Since we do not know for a fact which measures will be relevant for our problem, we perform two rounds of training: the first one to calculate the importance of each feature, and a second one with the most relevant measures, which will output the decision trees that comprise our switch criteria. The statistical measures empirically chosen were thought out to convey how the prediction varies throughout timesteps. They are:

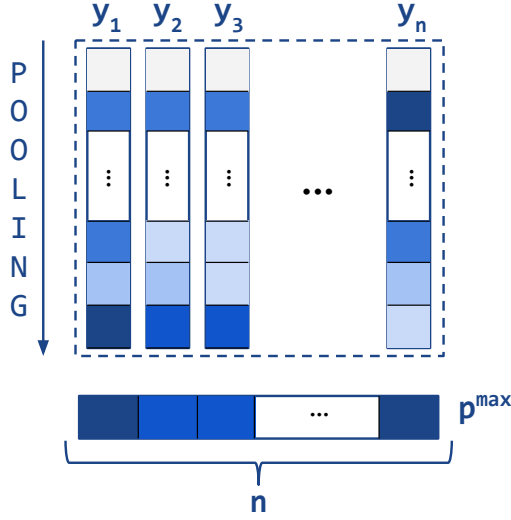


Figure 4.6: Composing vector p^{max} of maximum activations from all timesteps. A unidimensional max pooling is applied to every y_t , extracting the highest probability value from each timestep.

- **Mean:** It measures the mean value of our vector to provide an average prediction strength. It is calculated as follows:

$$mean = \frac{\sum_{t=1}^n p_t^{max}}{n} \quad (4.8)$$

with n being the number of ROI from the input scene.

- **Maximum Score:** Given that the mean can be misleading, we also added a measure of the strongest prediction on our vector, given by:

$$score = \max p^{max}. \quad (4.9)$$

- **Unbiased Variance:** It is a measure of how spread is the distribution. Since the naive variance yields a biased estimation, we use its unbiased version, where the size of our vector (n) minus one is set as the denominator, as showed by Equation 4.10,

$$variance = \frac{\sum_{t=1}^n (p_t^{max} - mean)^2}{n - 1}. \quad (4.10)$$

- **Kurtosis** [Pearson, 1905]: It describes the shape of a given input distribution, measuring its "tail". Heavy tails (or large kurtosis) means the distribution contains outliers, while light tails is the exact opposite, i.e., infrequent extreme deviations. This measure was chosen to provide information regarding the prediction agreement between timesteps. Its equation follows:

$$kurtosis = \frac{\sum_{t=1}^n (p_t^{max} - mean)^4 / n}{\sigma^4} \quad (4.11)$$

with σ representing the standard deviation of p^{max} .

- **Skewness** [Pearson, 1894]: Its goal is also to describe the shape of a distribution, only this time measuring its lack of symmetry, i.e., how similar it looks to the left and right of the center point. It is also referred as Fisher-Pearson coefficient of skewness, and can be defined as:

$$skewness = \frac{\sum_{t=1}^n (p_t^{max} - mean)^3 / n}{\sigma^3}. \quad (4.12)$$

The unknowns have the same interpretation as previously described.

- **Number of Observations**: The size of our vector p^{max} varies depending on the amount of patches selected from the input scene, and it determines the number of recurrent iterations on our model. This measure is defined by:

$$nobs = n. \quad (4.13)$$

- **Coefficient of Variation**: It measures the dispersion of a given distribution, also referred as a relative standard deviation as shown by Equation 4.14,

$$variation = \frac{\sigma}{mean}. \quad (4.14)$$

The first round of training consists in generating multiple binary decision trees trained on random sub-samples of our set. Then, for each statistical measure, the average decrease in node impurity is calculated. It is also referred as *gini importance* [Gini, 1912], and it is calculated as the average decrease in impurity weighted by the proportion of samples that reach the node. Afterwards, the impurity decrease from every node that uses the given measure in all trees are averaged, outputting the final importance of that measure. For a better understanding, first let us see how to calculate the importance of node k (ni_k) on Equation 4.15,

$$ni_k = w_k C_k - (w_k^{left} C_k^{left} + w_k^{right} C_k^{right}) \quad (4.15)$$

with w_k representing the proportion of samples that reach node k , while *left* and *right* indicate the two children of a binary node. C_k is the actual impurity of node k , given by $\sum_{i=1}^N f_i(1 - f_i)$ with f_i being the frequency of label i at the node for all N classes. In our case, the trees are trained for a binary problem, indicating whether the input measures are from an accurate prediction or a misclassification.

Knowing the importance of each node, the feature importance is simply the ratio of importance from all nodes that use measure m , let the number of such nodes be called k_m , relative to the importance from all nodes (k_{all}). The feature importance is calculated as follows,

$$fi_m = \frac{\sum_{k=1}^{k_m} ni_k}{\sum_{j=1}^{k_{all}} ni_j}, \quad (4.16)$$

keep in mind that for each measure m we consider the interval $[1, \dots, k_m]$ on the sum to be composed only of nodes that use metric m . From that we can normalize the value of each feature importance dividing it by the sum of all f_i .

The measures with highest feature importance are then used at the second round of training, dismissing the other features. We maintain the training procedure, with every sample labeled as $\{0, 1\}$ respectively indicating a correct prediction and a misclassification of our approach. The output decision trees will then be used at test time as our switch criteria, evaluating the outputs for a given image by the likelihood of it being a reliable prediction. If the output is classified as potentially a correct prediction then $y' = y'_{our}$. On the other hand, if it is considered to be a weak inference, the final prediction is provided by the paired classifier, i.e, $y' = y'_{paired}$. By proposing the ensemble, we expect to improve classification performance over each paired classifier.

Chapter 5

Experiments and Results

5.1 Datasets

We evaluated our approach on three datasets widely known as benchmark for scene recognition, namely Scene15 [Fei-Fei and Perona, 2005], MIT67 [Quattoni and Torralba, 2009] and SUN397 [Xiao et al., 2010]. This section outlines in details the characteristics of each one.

Scene15 is a small dataset, compared to the MIT67 and SUN397 used in our experiments. It is composed of 15 classes of indoor and outdoor environments, which are all illustrated on Figure 5.1. The dataset was gradually built from 2001 to 2006 and can be attributed to three different references. The first 8 categories, with their names starred in Figure 5.1, were collected by Oliva and Torralba [2001] comprising outdoor categories. Later, Fei-Fei and Perona [2005] added 5 new classes: *office*, *kitchen*, *living room*, *bedroom*, *suburb*. After this addition, Scene15 became suitable for the problem of indoor class recognition. Finally, in 2006, Lazebnik et al. [2006] contributed with the classes *industrial* and *store*. At the time, most methods tested on Scene15 were based on handcrafted features and classic machine learning approaches. Since the rise of deep learning models, the classification performance increased rapidly on that dataset, reaching up to 95% [Nascimento et al., 2017], but it still presented as a test subject for most scene classification approaches.

After analyzing the behaviour of scene recognition methods on Scene15, Quattoni and Torralba [2009] observed that indoor scenes present a much greater challenge compared to outdoor scenes. That discovery led the authors to a great contribution, a dataset solely focused on the problem of indoor scene recognition, called MIT67. As the name implies, it is composed of 67 classes of a wide variety of indoor environments, conveniently organized in five main categories to facilitate visualization, as showed in Figure 5.2. Besides the greater difficulty of indoor scenes, the larger number of classes causes MIT67 to be a bigger challenge than Scene15, which is reflected by the average classification performance of most methods, which is over 87% [Nascimento et al., 2017].

With the goal of taking scene classification to the next level and mainly motivated by the gap in size of object datasets compared to scene datasets, Xiao et al. [2010]



Figure 5.1: Samples from each class of Scene15. Image from Lazebnik et al. [2006].

introduced SUN397, a dataset of indoor and outdoor environments comprising 397 main classes, with a current hierarchical organization that amounts to a total of 908 total categories. For instance, the lowest hierarchical level of the class airport is divided into *airport airport*, *airport entrance*, *airport terminal* and *airport ticket counter*. As most methods in literature, for our experiments we consider only the 397 main categories, but it is worth highlighting the author’s goal to produce a dataset as complete as possible, even providing object categories for each scene category and human performance on scene recognition. The construction of such a large dataset, as highlighted by the authors, became viable due to the advancement of search engines with a better response for research queries. As for classification performance of most methods in the literature, SUN397 is by far the most challenging compared to the other two showed here, which can be attributed to its attempt to capture the full variety of scene classes.



Figure 5.2: Samples from MIT67 divided into 5 main categories for better visualization of the broad variety of scene classes [Quattoni and Torralba, 2009].

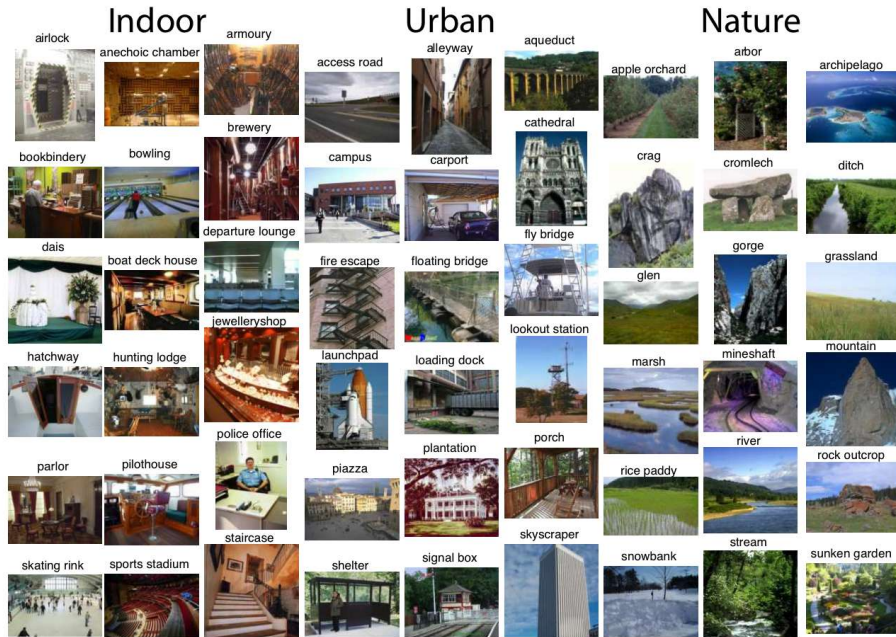


Figure 5.3: Samples from SUN397, divided into 3 major categories: indoor, urban and nature [Xiao et al., 2010].

5.2 Scenes as Sequences of Objects

In order to transform a single image into a sequence of object features, we exploit the power of Selective Search, a region proposal approach, to provide us with parts of the image that are most likely to contain objects. Selective search mainly requires two parameters: σ , which will feed a Gaussian filter to smooth the image before computing edges, and k corresponding to a scale parameter. Larger k values will prioritize larger components, and the opposite is also true for a smaller k in cases where attention to details is important. We used the default parameters proposed by [Felzenszwalb and Huttenlocher, 2004], which provides the starting locations for Selective Search. The values are $\sigma = 0.8$ and $k = 300$.

As discussed earlier, the number of ROI proposed by Selective Search for each scene can reach hundreds or even thousands, making it difficult for proper training on our available infrastructure. Hence, we proposed a filtering approach. Our main concern is to reach a balance between decreasing the number of ROI per image but maintaining significant coverage of image area, such that it does not discard the available information present at the scene. To validate our approach of representing an image as a sequence we used two datasets: Scene15 and MIT67.

The hyperparameters t_{lower} and t_{upper} , representing the thresholds of patch size relative to image size, were empirically set to 0.1 and 0.8, i.e., patches that account for less than 10% or more than 80% of the image area were discarded. That choice of parameters was based on the fact that objects can exist at different scales, but they usually tend to be smaller. We are aware that such parameters might require optimization, however the intuitive choice led to the following results, which meet the

criteria of balance between sequence length and image coverage.

First, we analyzed the average number of patches per class before and after filtering. Figure 5.4 shows the results for Scene15, where we can see the drastic decrease in number of patches, which after filtering is much lower than a hundred for all classes. The exorbitant decrease is mainly due to the large amount of tiny patches proposed by Selective Search. The same results for MIT67 are presented in Figure 5.5. It is consistent with Scene15, showing a major decrease in proposed ROI.

A few additional notes on that result is the perception that the number of patches can also convey information regarding scene category. For instance, on Scene15, class *coast* shows one of the least amount of patches selected since it does not usually have much object information present at the scene. Meanwhile classes such as *store*, *forest*, and *inside city* are by far the highest, meaning they are usually more crowded, either of outdoor objects such as trees and buildings, or merchandise and commodities at stores. For MIT67 the necessity of filtering is even higher, since classes like *grocery store* are represented by over 1,500 patches per image.

Certainly, our approach raises the question of how much information is being discarded. Essentially the filtering proposal relied on the fact that Selective Search is a hierarchical approach, joining segmented patches from the entire image at higher scales at every iteration. That leads to the assumption that patches within an intermediate range will account for a large percentage of the image. Specially considering that our intermediate range is defined at both extremes of the scale. To make sure the assumption holds, for each image we defined a single polygon as a junction of all selected ROI after filtering, as presented in Figure 5.6. The ratio between the polygon area and image area determined a coverage percentage of all proposed regions. Figure 5.6 is a real example of patches selected after filtering, and its joint polygon of patches accounts for 82.72% of the image.

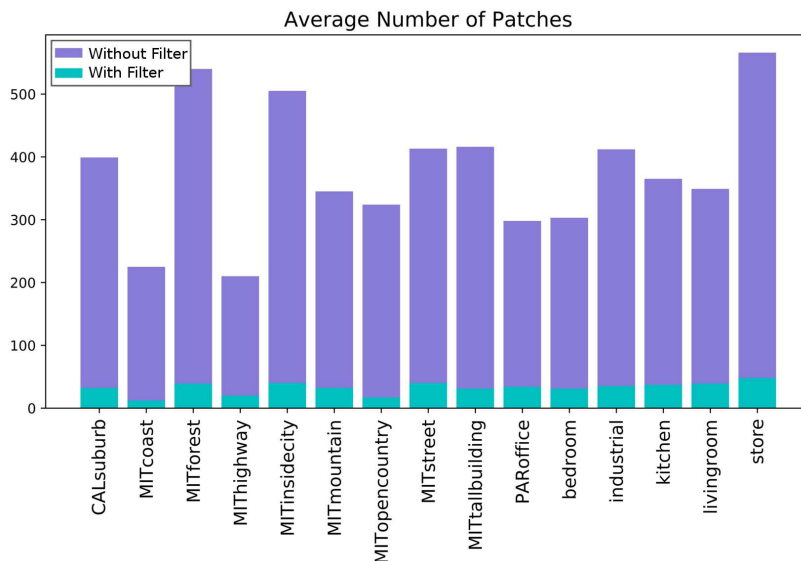


Figure 5.4: Average number of patches proposed by Selective Search before and after filtering for Scene15

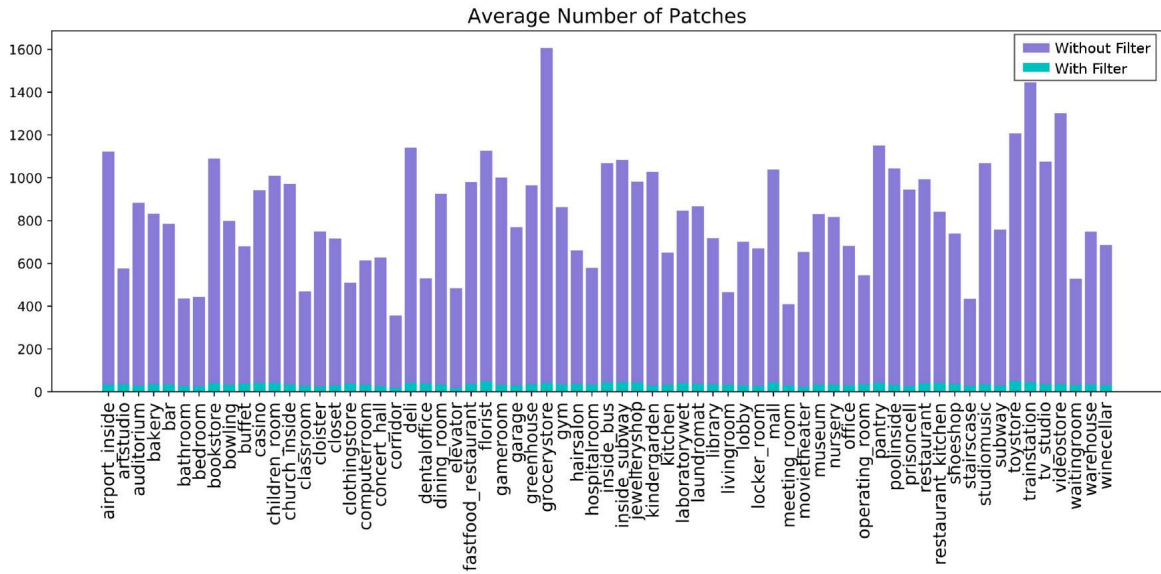


Figure 5.5: Average number of patches proposed by Selective Search before and after filtering for MIT67

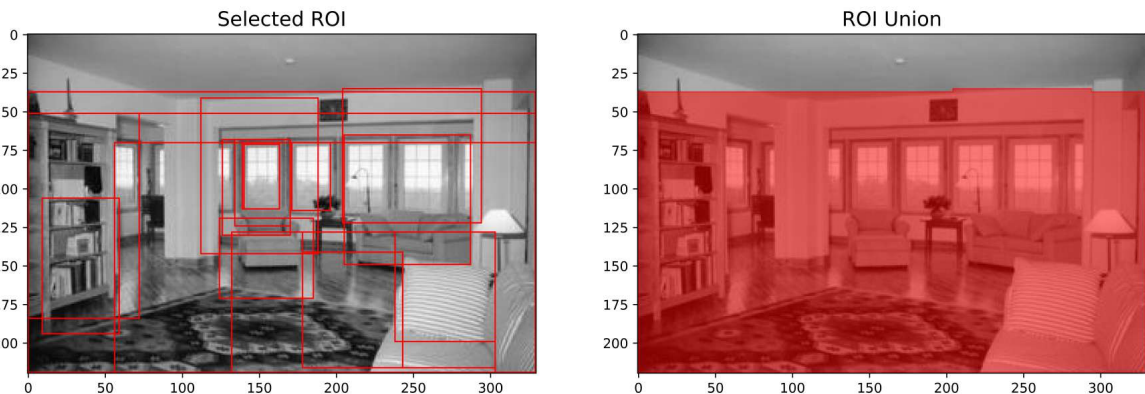


Figure 5.6: Example of coverage percentage analysis for a single image. All the proposed ROI (left) was joined into a single geometric shape (right).

After calculating the coverage percentage for all images, we separated them by class, and averaged the values for each class. For MIT67, results are shown in figure 5.7. The proposed regions still cover over 90% of the image area for all classes, but mostly it reaches the maximum coverage at 100%. Scene15 results are also positive, as shown by Figure 5.8, staying above 90%. That result is highly relevant since the characteristics of Scene15 is usually providing cleaner environments such that it does not cause any clutter or major occlusions, which could have affected the image coverage after discarding patches.

Once we have the proposed ROI after filtering, the last step to compose a sequence is to extract features from each patch, maintaining its original order on the final sequence. We leverage high quality features from the final convolutional layer of Resnet-50 pretrained on Imagenet. Feature extraction follows the pre-processing

protocol of most CNN references, down-sampling the image to a fixed size (224×224) and a per-pixel subtraction by the training mean activity [Krizhevsky et al., 2012; He et al., 2016].

5.3 LSTM Settings

As outlined in Chapter 4, our recurrent model follows a M2M training approach with a BiLSTM followed by a fully connected layer and a softmax activation. The input that feeds our BiLSTM has three dimensions: $batch_size \times seq_len \times feat_size$, representing respectively the batch size, sequence length and feature size. The first parameter was fixed to 1 to avoid the need to pad our data, since seq_len varies per sample with the amount of selected ROI. $feat_size$ is determined by the network we chose, Resnet-50, which outputs a vector with 2048 dimensions. There is also a free parameter on the recurrent layer concerning its hidden size, i.e., the size of its output h_t (refer to Equation 2.3). Considering that we tested different architectures and training approaches, as it will later be showed on Section 5.7, h_t was fixed to 512 as proposed by the work of Javed and Nelakanti [2017]. It is worth reminding that our recurrent layer is bidirectional, which means that although $h_t = 512$, the actual outputs is $2 \times h_t$ since the output of both recurrent units (one for each direction) will be concatenated before feeding the next layer.

Seeing that we want an output for each input (synchronized M2M), the recurrent output from each timestep t will be forwarded through the fully connected layer. Hence, its input has dimensions $1,024 \times n_c$ depending on the number of classes n_c of the training dataset.

As for training settings, we used Adam [Kingma and Ba, 2014] with its default parameters, except for its initial learning rate that was empirically set as $1e - 7$. The

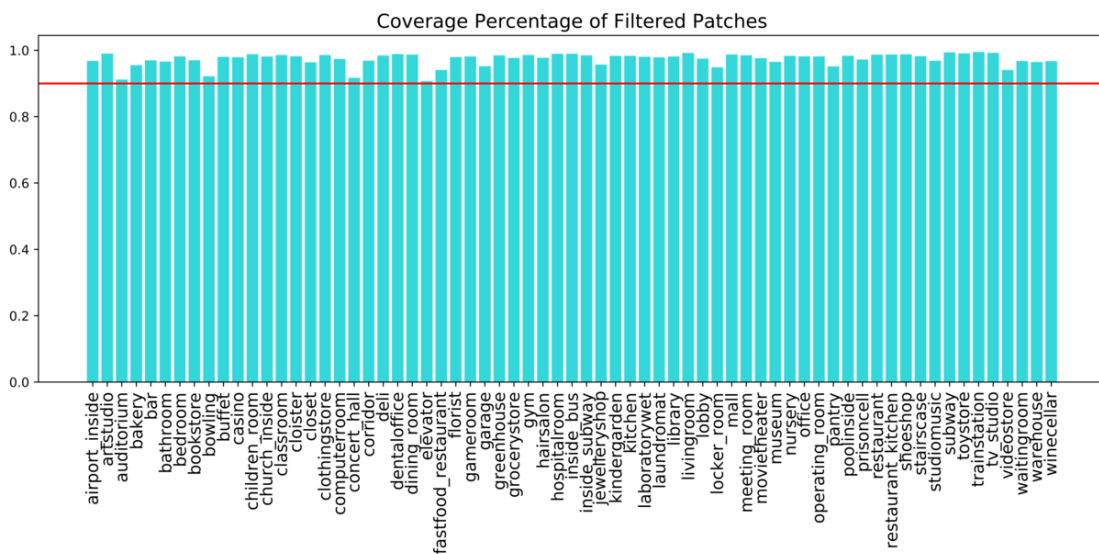


Figure 5.7: Average coverage percentage of patches from MIT67 after filtering. The red horizontal line marks the percentage 0.9.

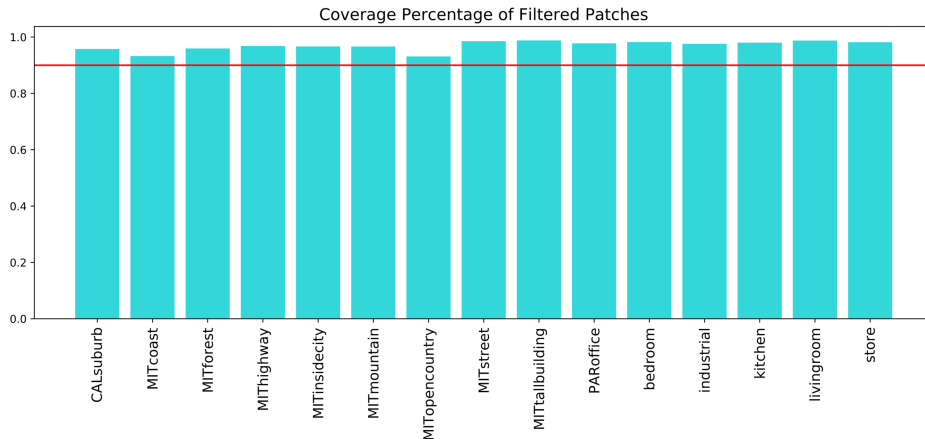


Figure 5.8: Average coverage percentage of patches from Scene15 after filtering. The red horizontal line marks the percentage 0.9.

train/test split is already defined on the reference of each dataset. However, we needed a validation set in order to generate our weight matrix W^{obj} and to train the switch criteria on the ensemble of classifiers. Considering that all training sets have around 80 to 100 samples for each class, we created a validation set for each dataset by randomly selecting 15 samples from each class, removing such samples from the training set.

5.4 Object Weights

One of the most important aspects of our proposition is a weighted majority voting based on a weight matrix that determines how each object category relates to a given scene. In order to understand the characteristics of our matrix, we will keep working with Scene15 and MIT67, since they are more manageable and due to the smaller amount of classes, it allows a visualization of per-class results.

As mentioned at Section 4.2, the size of the weight matrix is $n_c \times 1,000$, representing respectively the number of scene categories and all object classes from Imagenet. It would not be practical or convenient to show the entire matrix, mainly due its size and sparsity. For Scene15, 82.90% of the cells are zero values, leaving a small percentage of meaningful objects per class, 170 on average, which agrees with an intuitive interpretation of how many objects can appear on each scene category. Figure 5.9 illustrates that behaviour by showing the amount of non-zero values on each row of our matrix built for Scene15. The plot was ordered increasingly to highlight the fact that, aside from classes bedroom and forest, indoor classes have a larger quantity of significant objects than outdoor scenes. That behaviour is very consistent with findings from Quattoni and Torralba [2009], stating that indoor scenes tend to be more crowded with a large variety of objects, which is one of the main reasons that pose indoor scene recognition as a greater challenge.

Given that Scene15 is a small dataset in number of classes, it is possible to partially visualize the characteristics of our matrix for all classes. Figure 5.10 shows the Top-5 object weights and its categories for each scene class. In other words, the

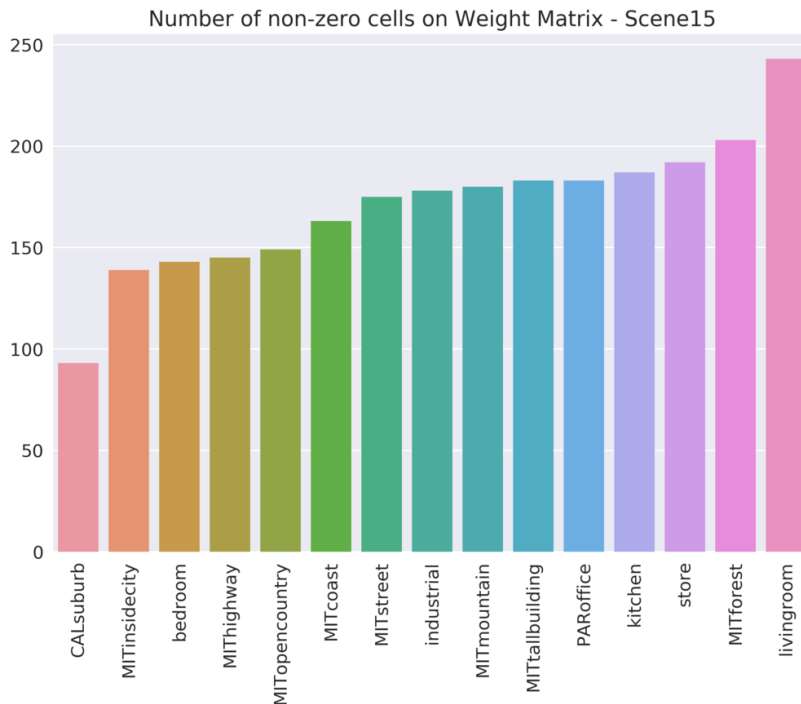


Figure 5.9: Bar plot representing the amount of non-zero cells on each row of our weight matrix for Scene15. On average, around 170 objects out of 1,000 candidates are meaningful for each scene according to our proposal.

plot shows the 5 greatest values on our weight matrix W^{obj} for the corresponding row i of each class. The number 5 is a choice merely based on the available space to provide a clean visualization. Since we are relying on predictions of object labels, the presence of objects that are semantically related to the respective class on its Top-5 is a positive result, meaning the weight matrix is attributing higher weights to the expected objects.

As a reminder, we rely on those weights to aggregate BiLSTM predictions from all timesteps, since they provide the relevance of each object to all classes. For a further understanding of how the weights contribute to the prediction, as defined by Equation 4.6, let Figure 5.11 illustrate the weight vectors for three objects from Scene15: window shade, moving van and fox squirrel, all of which occur in very different scenarios. Given our matrix W^{obj} , each row i corresponds to a scene category, and each column j represents an object class. Given an object prediction j from an input patch, we select the corresponding column in our matrix, which provides a vector of size n_c . Figure 5.11 presents such vectors, highlighting their differences. For instance, the presence of a window shade will enforce a prediction towards class living-room, while applying lower weights for any other class. The same goes for object moving van towards street scenes, and a fox squirrel in forest scenes.

For MIT67 the percentage of zero-value cells is 86.20%, which can be inferred by Figure 5.12, showing on average 138 significant objects out of 1000 candidates from Imagenet. It is possible to visualize on that plot the variability between classes, for instance class *poolinside* shows one of the smallest amounts of meaningful objects. The same class was previously used as an illustration of region proposals on classes with

CALsuburb	[mobile home, Kerry blue terrier, steel arch bridge, boathouse, fire screen]	[0.164, 0.108, 0.053, 0.049, 0.044]
MITcoast	[seashore, drilling platform, breakwater, promontory, airship]	[0.11, 0.078, 0.061, 0.06, 0.06]
MITforest	[chain mail, Kerry blue terrier, park bench, steel arch bridge, fox squirrel]	[0.113, 0.105, 0.047, 0.045, 0.033]
MIThighway	[steel arch bridge, trailer truck, passenger car, fire screen, airship]	[0.133, 0.121, 0.073, 0.053, 0.049]
MITinsidicity	[fire screen, prison, palace, tobacco shop, cinema]	[0.11, 0.1, 0.056, 0.054, 0.037]
MITmountain	[alp, grey whale, volcano, Kerry blue terrier, airship]	[0.3, 0.053, 0.037, 0.03, 0.027]
MITopencountry	[Kerry blue terrier, chain mail, airship, steel arch bridge, hay]	[0.07, 0.066, 0.051, 0.041, 0.034]
MITstreet	[prison, streetcar, gondola, cab, steel arch bridge]	[0.151, 0.089, 0.07, 0.063, 0.053]
MITtallbuilding	[airship, fire screen, prison, radiator, space heater]	[0.242, 0.051, 0.051, 0.044, 0.032]
PARoffice	[fire screen, desk, library, screen, photocopier]	[0.08, 0.059, 0.055, 0.055, 0.045]
bedroom	[studio couch, fire screen, four-poster, quilt, window shade]	[0.208, 0.087, 0.084, 0.042, 0.036]
industrial	[drilling platform, steel arch bridge, fire screen, prison, airship]	[0.121, 0.091, 0.05, 0.043, 0.039]
kitchen	[microwave, fire screen, espresso maker, barbershop, dishwasher]	[0.13, 0.075, 0.052, 0.045, 0.038]
livingroom	[fire screen, studio couch, window shade, prison, tobacco shop]	[0.098, 0.087, 0.056, 0.033, 0.031]
store	[tobacco shop, grocery store, fire screen, bulletproof vest, espresso maker]	[0.254, 0.03, 0.029, 0.026, 0.025]

Figure 5.10: Top-5 object weights for all classes of Scene15. Column one has the name of each scene class, column two shows the categories of the matrix’s Top-5 objects, followed by their respective weights on the last column.

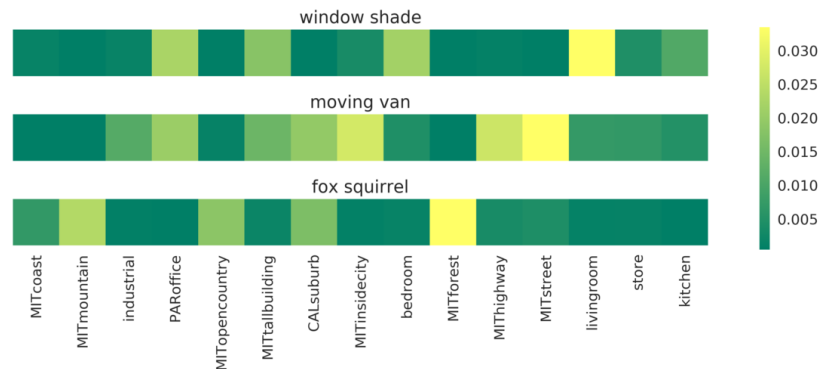


Figure 5.11: Weight vectors from matrix W^{obj} at the columns corresponding to three different object categories. The colors represent the strength of each object weight towards all scene classes.

fewer objects (refer to Figure 4.2), while *florist* is by far the highest, given the diversity of objects on such category.

Even though it would be impractical to present the Top-5 objects for all classes on MIT67, it is still necessary to visualize the matrix in order to understand the effectiveness of our voting approach. Let us look at classes we are most familiar. Figure 5.13 shows the top 30 weights for classes bathroom, bedroom, living-room, and kitchen. In other words, the plot shows the 30 greatest values on our weight matrix W^{obj} for the corresponding row i of each class. The value 30 was an empirical choice, based on the point where weights are significantly lower, tending to zero. The x-axis labels of those

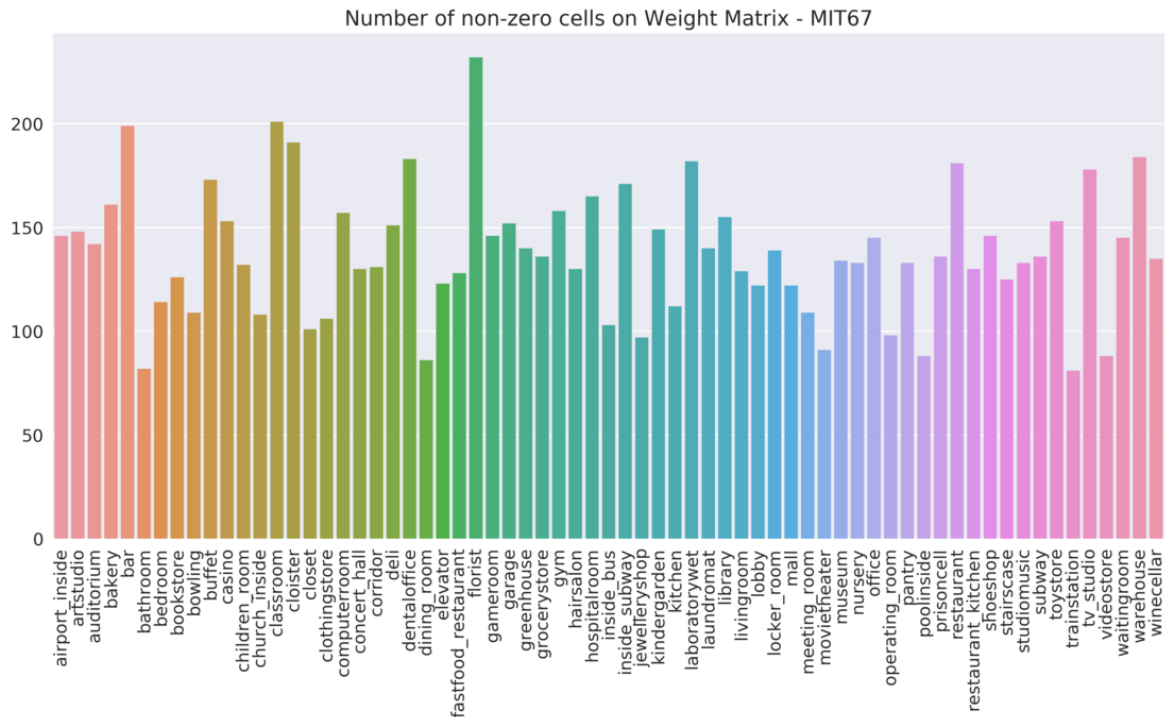


Figure 5.12: Bar plot representing the amount of non-zero cells on each row of our weight matrix for MIT67. On average, around 138 objects out of 1,000 candidates are meaningful for each scene according to our proposal.

plots also show the names of object classes corresponding to each weight, so that we can visualize which objects are chosen as worthy of greater importance during voting.

The results roughly illustrated by Figure 5.13 are a good approximation of human expectations on which objects are more representative for a scene category. For instance, class bathroom shows greater importance for items such as wash basin, bathtub, toilet seat, etc. While the Top-2 objects of a kitchen on our matrix is microwave and dishwasher. An interesting behaviour is how classes bedroom and living-room share quite a few objects and at the same time have important distinctions. Since Imagenet does not have a specific category for beds, its occurrences are labeled as studio couch, also referred as day bed. It is noticeable that the same object (day bed) can have a different importance depending on the scene category, and the methodology must rely on the given context for an accurate prediction when it sees a day bed equivalent. As a reminder, the datasets used in this Thesis do not have object labels, so the classes presented in Figure 5.13 are predictions from Resnet-50. Although those are very high quality predictions, there are misclassifications such as class firescreen, it stands out as significant for most classes when it is in fact not present.

The main experiment to validate the importance of our object matrix are present at Table 5.1. We compared a vanilla majority voting against our weighted proposal on all three datasets (Scene15, MIT67 and SUN397). For Scene15, the improvement in performance is small but still valuable. Since we are directly interfering with the probability vector, there was a chance to decrease the performance, which did not

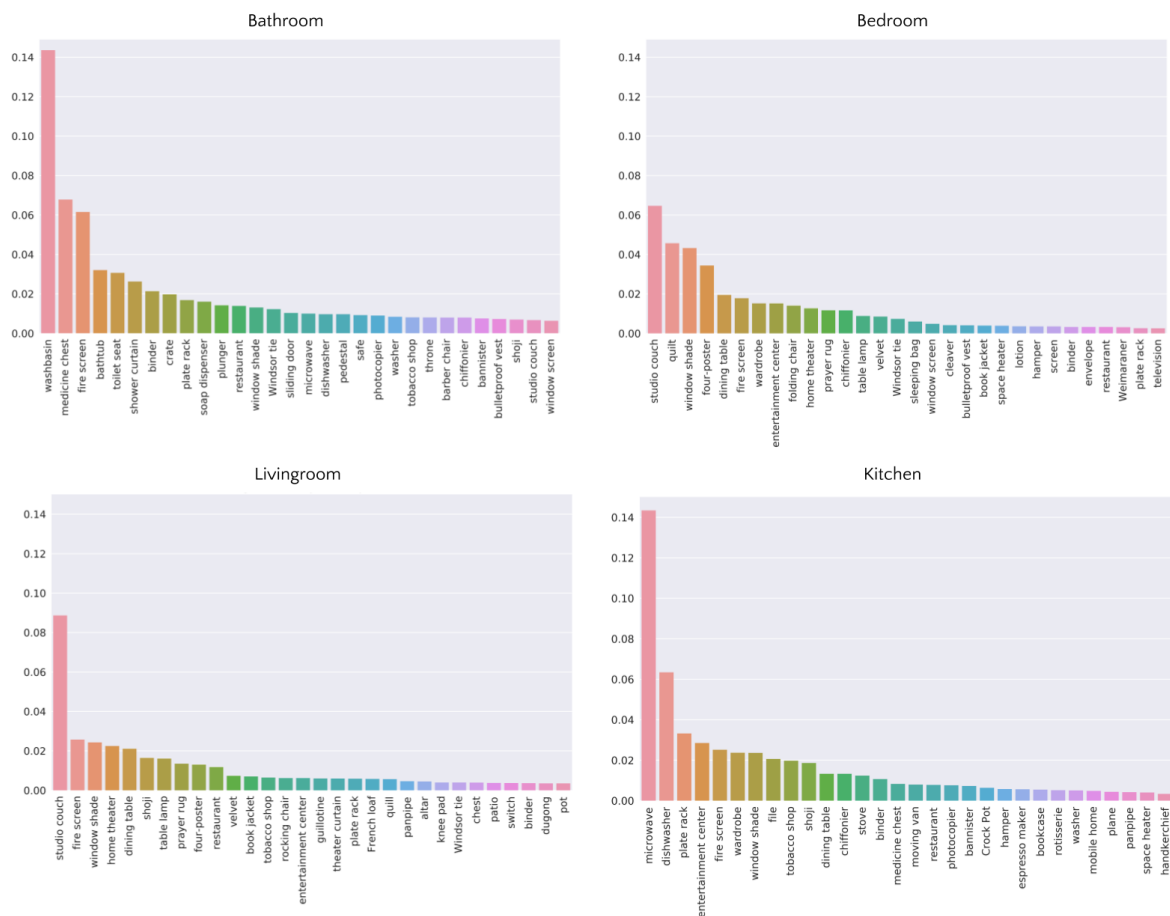


Figure 5.13: Top 30 object weights for bathroom, bedroom, living-room and kitchen on MIT67.

	Scene15	MIT67	SUN397
Majority Voting	94.06%	75.18%	51.26%
Weighted Majority Voting	94.29%	79.52%	54.00%

Table 5.1: Results with vanilla majority voting compared to our weighted approach.

happen in any case we tested. On the other hand both MIT67 and SUN397 showed significant gain, respectively 4.34 and 2.74 percentage points. It is worth noting the gain for SUN397, given its size and notorious difficulty.

5.5 Ensemble of Classifiers

According to the proposal of our ensemble, the first step is to train randomly generated trees in order to calculate the feature importance of each empirically proposed measure. We used the default number of trees from Python’s Random Forest Scikit-Learn

library [Pedregosa et al., 2011], which is 100. The Random Forest approach [Breiman, 2001] will perform as described at the methodology, generating multiple trees on subsamples of our data, to avoid overfitting with a single tree. The maximum depth of each tree was empirically set to 5, as an additional effort to avoid building large and very specialized trees. Figure 5.14 shows the feature importance results for each statistical measure on all datasets. To understand each metric please refer to Section 4.4. There is a considerable level of agreement for all datasets towards measures of mean and variation. Both will provide information regarding the certainty of predictions. Those are very intuitive measure choices, while high means indicate overall strong activations, small variations can be interpreted as agreement between timesteps.

The second round of training counts only on measures of mean and variation, building once again a total of 100 trees as part of a Random Forest, and training them on the validation set. As a reminder, it is a binary problem to classify our method’s outputs and determine if our prediction should remain as final output or if the paired approach should be called. Once the training is done, the output Random Forest serves as the final switch criteria. Figure 5.15 shows one of the many randomly generated trees to illustrate the decision making process of our ensemble. Decision trees allow a interpretable understanding of the switch criteria. For instance, on Figure 5.15, the threshold $mean \leq 0.911$ on the root node appears to be highly relevant, separating with notable quality the two classes. That visualization was provided by a module entitled *tree* from Scikit-Learn. It prints four rows for each node, representing respectively the feature that splits the node and its threshold; the gini importance, i.e., node impurity; number of samples that reach the node; and finally the frequency of each label.

Three methods from the literature were chosen to act as paired classifiers for our ensemble. They were chosen either for providing a source code or a trained model. First, a VGG16 [Simonyan and Zisserman, 2014] pretrained on Places [Zhou et al., 2014a] was fine-tuned as proposed by Nascimento et al. [2017]. The second method

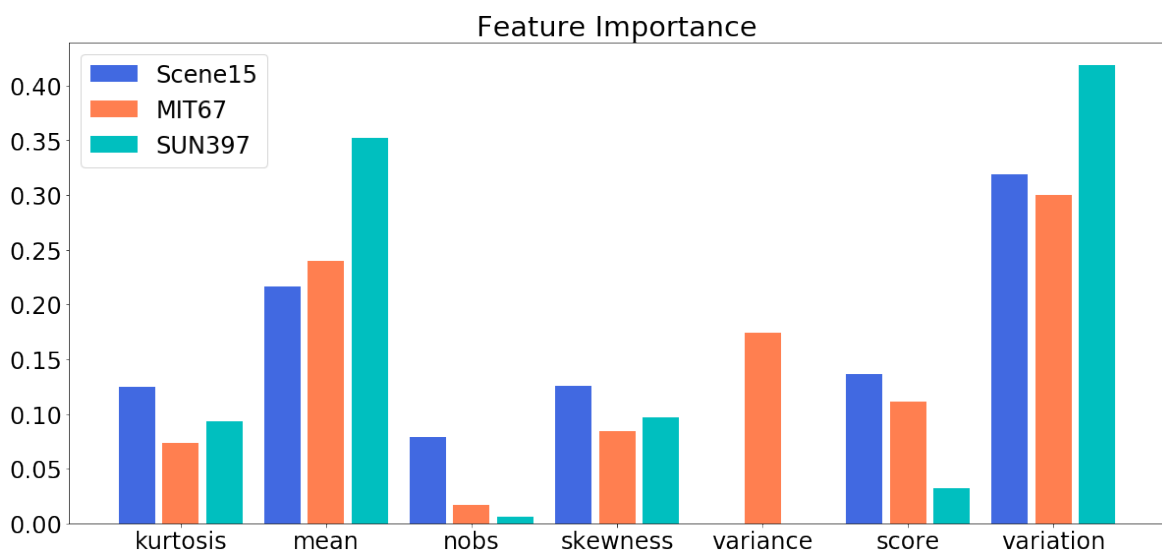


Figure 5.14: Feature importance results for each empirically selected statistical measure. Results for all three datasets.

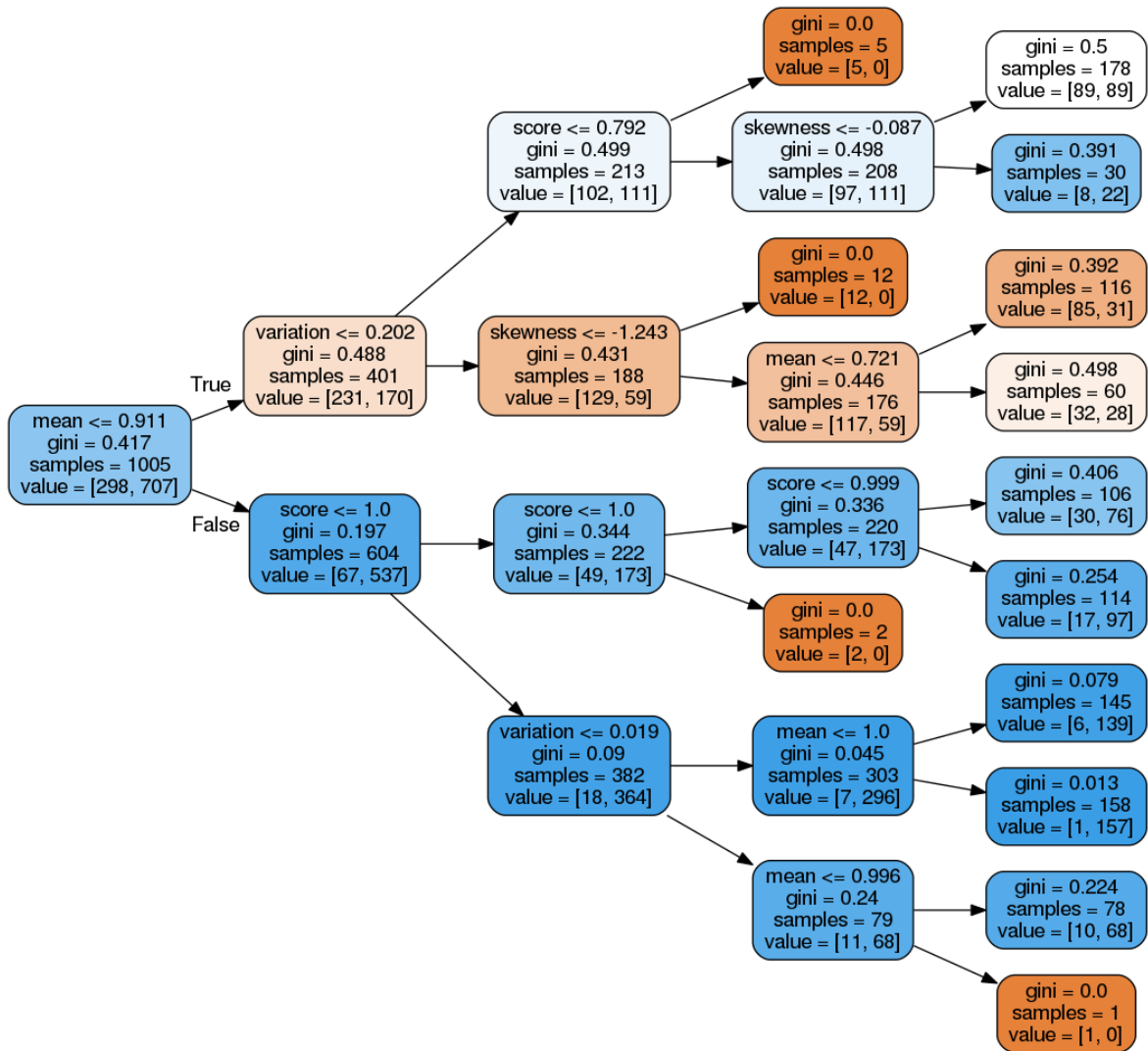


Figure 5.15: Random decision tree from the output Random Forest. Orange leaves are labeled as misclassifications and blue leaves indicate our method will provide the final output. Intensity of colours shows the node impurity (gini importance).

was Herranz et al. [2016], a method that follows the same premise as ours regarding the relevance of objects, even though it is based on a CNN approach. Finally, Nascimento et al. [2017], one of the best performances on the literature, also proposing a CNN based methodology. Table 5.2 shows our results with the proposed M2M BiLSTM with a weighted voting but without any ensemble, and compares it to each of the paired methods by themselves and as part of our ensemble. We improved the classification accuracy over each method, specially VGG-Places, an approach entirely based on global features, compared to which we gained 0.53, 3.72 and 1.18 percentage points on Scene15, MIT67 and SUN397. We find quite relevant that the best improvement happened on a dataset dedicated to indoor scenes (MIT67), since our work was modeled towards that problem. For Herranz et al. [2016] and Nascimento et al. [2017] although the performance had little increase, it is a very significant result considering

	Scene15	MIT67	SUN397
M2M BiLSTM Weighted Voting	94.29%	79.52%	54.00%
VGG16 (Places) Ensemble (VGG16)	93.87% 94.40%	80.88% 84.60%	66.90% 68.08%
Herranz et al. [2016] Ensemble (Herranz)	95.18% 95.96%	86.04% 86.47%	70.17% 71.35%
Nascimento et al. [2017] Ensemble (Nascimento)	95.73% 96.30%	87.22% 88.25%	71.08% 71.81%

Table 5.2: Accuracy results for the ensemble of paired classifiers. Our method was paired with three literature approaches, improving over each of them.

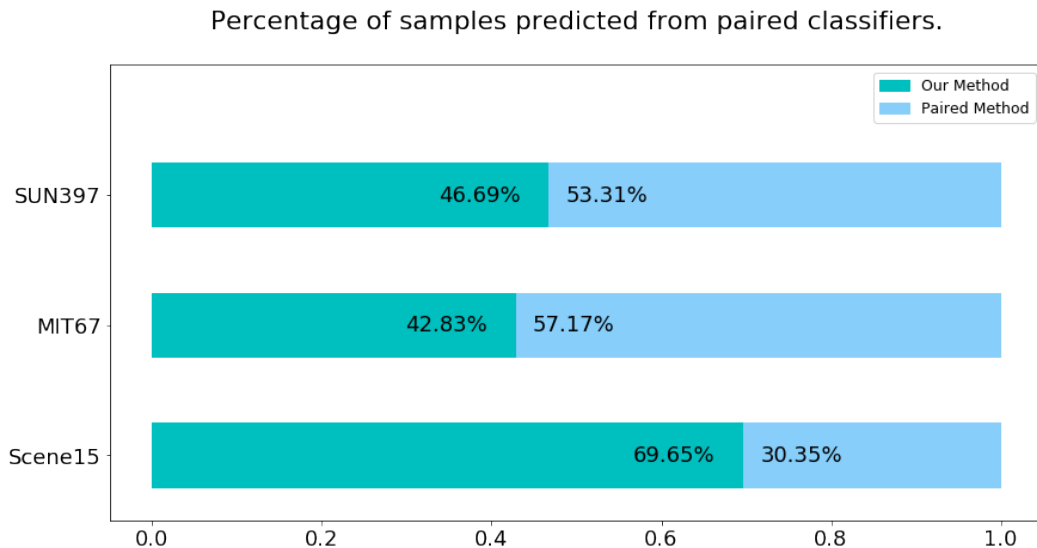


Figure 5.16: Percentage predicted by our method and by paired classifiers according to the switch criteria.

that we are improving over very successful methods from the literature.

One question worth raising is which percentage of samples our method was responsible for classifying. It is important to know how dependent we are of paired approaches in order to achieve competitive results. Figure 5.16 answers that question by showing the percentages for each dataset. Seeing that the switch criteria is trained per dataset and applied only to our method, the paired approach is inconsequential with respect to the ensemble decision making. On average, we predicted over 53% of samples, serving as evidence that our approach has a significant role on the ensemble. That result can be interpreted as how distinguished is the behavior of our method when it outputs an accurate classification compared to misclassifications.

5.6 State-of-the-Art Scene Recognition

On this section we compare our work to state-of-the-art approaches, presenting works of two different natures: CNN-based and RNN-based. Table 5.3 shows the accuracy of all methods including the two possible outputs produced by our proposal: M2M BiLSTM with a weighted majority voting, and the best result with an ensemble of paired classifiers presented on Table 5.2, which for all cases is by pairing with Nascimento et al. [2017].

Our method, even without the ensemble, performs better than any other RNN-based approach on MIT67, which is an entirely indoor dataset. That result is very positive since all approaches rely on the same premise of correlating interdependent image parts, with Wang and Pan [2017] basing its methodology on the same fundamental premises as ours with respect to sequence composition from scene images. None of the RNN approaches presented results for Scene15, but they did for SUN397, which showed interesting results in comparison to ours. Specially the work of Zuo et al. [2016], presented twice at Table 5.3, which pretrains its model on two different datasets: ImageNet (ILSVRC), with object-centric samples, and Places, a large scale dataset for scene recognition. Our performance is better than its ILSVRC variation for both datasets (MIT67 and SUN397), whereas the scene-centric pretraining beats our accuracy on SUN397 by a large margin. From that, we can infer that since SUN397 has over half of its samples dedicated to outdoor scenes (55.41%), a methodology based on correlation of object parts has little capacity to compete with features that encode global structures. That result triggers the necessity of evaluating all methods only on indoor samples, since as defended by Quattoni and Torralba [2009] it is a more challenging classification problem that requires approaches specially tailored to it, as the one proposed by this Thesis. The information on indoor samples of mixed datasets is not available for any of those RNN-based approaches.

As for CNN-based approaches, Table 5.3 starts by presenting the performance of an SVM classifier, with its default parameters from Scikit-Learn, trained and tested with features from a Resnet-50 pretrained on object images (ILSVRC), since it is the feature extractor used by our methodology. We outperform it without any ensemble, indicating the level of improvement added by our proposal. Even when Resnet-50 is pretrained with a dataset from a closer domain (Places), also feeding the SVM classifier, our approach still presents a better performance on both Scene15 and MIT67. As for SUN397, scene-centric features seem to have higher quality than correlating object features. This result indicates that correlating local information based solely local object features can be just as valuable as a deep CNN pretrained on large scale scene-centric data, specially for indoor scenes.

		Scene15	MIT67	SUN397
CNN-based	Resnet-50 (ILSVRC)	90.87%	69.13%	53.70%
	Resnet-50 (Places)	92.03%	74.73%	60.33%
	VGG16 (Places)	93.87%	80.88%	66.90%
	Herranz et al. [2016]	95.18%	86.04%	70.17%
	Wang et al. [2017]	-	86.20%	73.00%
	Nascimento et al. [2017]	95.73%	87.22%	71.08%
RNN-based	Zuo et al. [2015]	-	65.07%	51.14%
	Zuo et al. [2016] (ILSVRC)	-	69.25%	52.78%
	Zuo et al. [2016] (Places)	-	75.67%	60.34%
	Wang and Pan [2017]	-	71.86%	57.72%
Our Method	M2M BiLSTM			
	Weighted Voting	94.29%	79.52%	54.00%
	Ensemble	96.30%	88.25%	71.81%

Table 5.3: Comparing the accuracy our proposed approach with methods from the literature. Results were separated by the main methodology nature: CNN-based and RNN-based.

Following, we show the performance of the baseline proposed by Nascimento et al. [2017], a VGG16 pretrained on Places and fine-tuned on SUN397, a dataset even closer to our explored domain. Even though it presents better results relative to raw Resnet-50 features, our proposal is still highly competitive on MIT67. As previously mentioned, we tailored our proposal for the problem of indoor scene classification, hence MIT67 provides the most valuable insight regarding the quality of our approach.

The remaining CNN-based approaches showed in Table 5.3 are more sophisticated methodologies from the literature, some of them used as paired classifiers on our ensemble. Their performance are outstanding on all three datasets. That could be attributed to the more extensive history of applying CNN approaches to the problem of scene recognition, allowing the field to grow on a fast pace throughout the years relative to RNN-based methods. Essentially, there is much yet to be researched on recurrent approaches, which rose after CNNs. Judging by the performance increase of RNN-based methods throughout the years, it is important to unravel the full potential of recurrent methods for image classification problems. By experimenting with CNN approaches as part of our ensemble, we found that there is still room for improvement on such methods that can be provided by high quality correlation of local information.

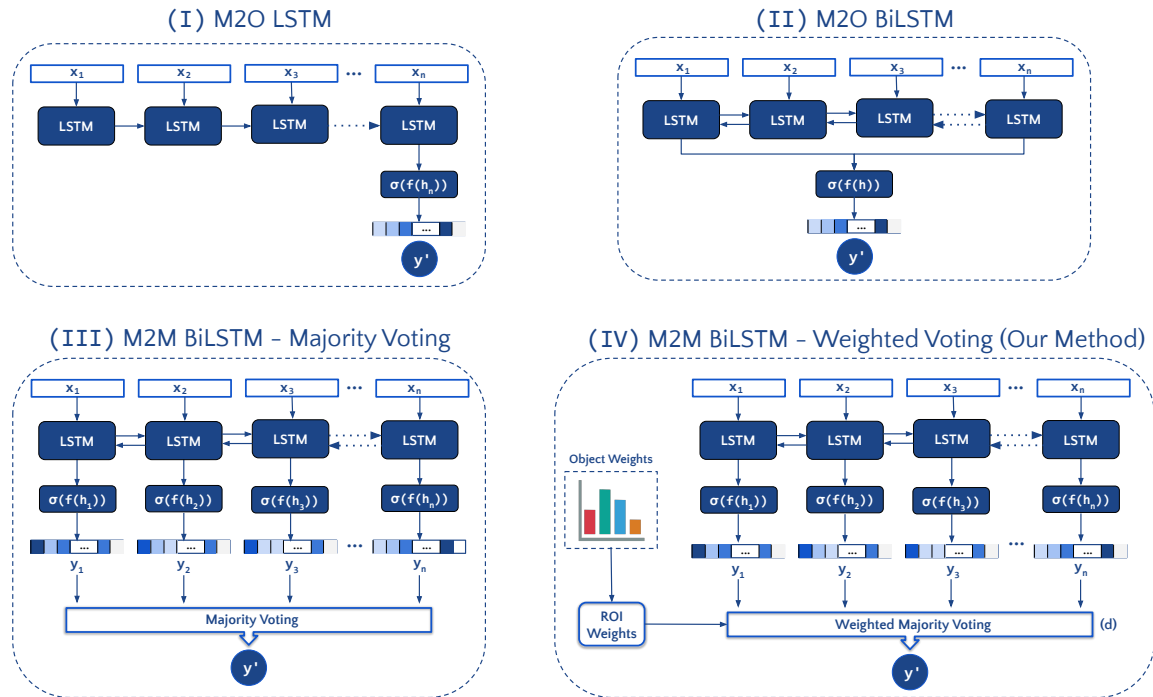


Figure 5.17: Variations of recurrent approaches for scene recognition in order to analyze the contribution of each aspect of our method. The most simple is a Many-to-One (M2O) unidirectional LSTM, followed by variation II, a M2O BiLSTM. Variation III is a M2M BiLSTM with vanilla majority voting, and finally our method adding a weighted majority voting as a late fusion of predictions.

5.7 Ablation Analysis

For the sake of understanding how we reached the proposed methodology, this section unpacks each individual aspect related to the context modeling through a recurrent approach, referring to steps (c) and (d) of our methodology (see Figure 4.1). That means steps (a) and (b) will remain fixed as described in the methodology section. The ablation will be performed once again on Scene15 and MIT67 in order to validate a consistent behavior on datasets with different characteristics. Figure 5.17 displays four different configurations that were tested by this work.

Firstly, variation I proposes a unidirectional LSTM optimized by a Many-to-One (M2O) training procedure, a vanilla approach when it comes to recurrent models used for classification. That approach takes into account only the knowledge accumulated from the positive direction, following the decreasing order of *objectness* defined by Selective Search. Then, variation II adds a small but quite significant change with a bidirectional LSTM, also trained with a M2O approach. In that case, the data from both ends of the recurrent units are concatenated and fed to the dense layer in order to generate a single prediction. Variations III and IV are very similar in their goal to optimize every timestep to recognize the scene category, which requires a M2M training procedure. While variation III uses a regular majority voting to perform the prediction, variation IV refers to our complete method, adding a weighted majority voting in which

	M2O		M2M	
	LSTM	BiLSTM	BiLSTM Majority Vot.	BiLSTM Weighted Vot.
Accuracy	92.00%	93.50%	94.06%	94.29%
Recall	92.17%	93.50%	94.15%	94.47%
Precision	92.31%	93.61%	94.29%	94.75%
F1 Score	92.23%	93.55%	94.19%	94.57%

Table 5.4: Comparing different training approaches and recurrent architectures for scene recognition on Scene15.

each patch is weighted by the semantic relevance of its containing object.

For this experiment, we trained each methodology presented in Figure 5.17, building a model for Scene15 and another for MIT67, following the training procedure suggested by [Fei-Fei and Perona, 2005; Quattoni and Torralba, 2009]. Table 5.4 shows the results for Scene15. Since the dataset offers little challenge compared to others, the simplest configuration already achieves good results, reaching over 92% on all evaluated criteria. As a consequence, the contribution of each variation offers only a slight performance gain. However it is still possible to notice how a BiLSTM is more powerful in this context, reaffirming the value of accumulating knowledge from more than one direction. And while there were small changes from variation II to III, adding the weighted voting on variation IV improved the classification performance, achieving over 94%. Even with a modest improvement between variations, it is noticeable that our approach grew on the right direction, exploiting the advantages of a recurrent model to convey high quality contextual information. Additionally, a M2M approach also allows the consideration of additional criteria when gathering all predictions.

Table 5.5 shows the same results for MIT67, only now much more evident by the gap of performance between each variation. The greatly improved behavior of a BiLSTM compared to the unidirectional equivalent is consistent with early findings in

	M2O		M2M	
	LSTM	BiLSTM	BiLSTM Majority Vot.	BiLSTM Weighted Vot.
Accuracy	59.66%	72.94%	75.18%	79.52%
Recall	59.51%	72.85%	75.20%	79.60%
Precision	47.90%	74.65%	76.09%	80.13%
F1 Score	53.07%	73.74%	75.64%	79.86%

Table 5.5: Comparing different training approaches and recurrent architectures for scene recognition on MIT67.

the literature regarding the benefits of accumulating knowledge from different directions whenever the problem allows it [Schuster and Paliwal, 1997]. More importantly, on variation IV, the positive results by weighting the predictions supports our claim that not every region of the image is equally relevant, which allows iterations with little informative features or even misleading elements to compromise classification performance if not considered by the methodology. In other words, for the problem of scene recognition a recurrent approach can improve by taking into account the relevance of each patch with respect to the scene. It is also worth reminding that MIT67 is the only dataset entirely dedicated to indoor scenes, which is the main goal of our work. The greater improvement of our approach on this dataset is a very positive achievement towards the recognition of indoor scenes using a recurrent approach.

Besides exploring different architectures and training procedures for recurrent models, the weighted voting proposed here presented as a valuable contribution of this Thesis. For a deeper understanding of how it affects the performance of our method, Figure 5.18 shows a comparison of per-class performance on MIT67 of the vanilla majority voting against our approach with weighted votes. The positive impact of weighting the predictions of each timestep is significant for most classes, especially for categories like movietheater and classroom presenting great improvement. A little decrease is also noticeable for less than 10% of classes, which overall affects very little the average performance.

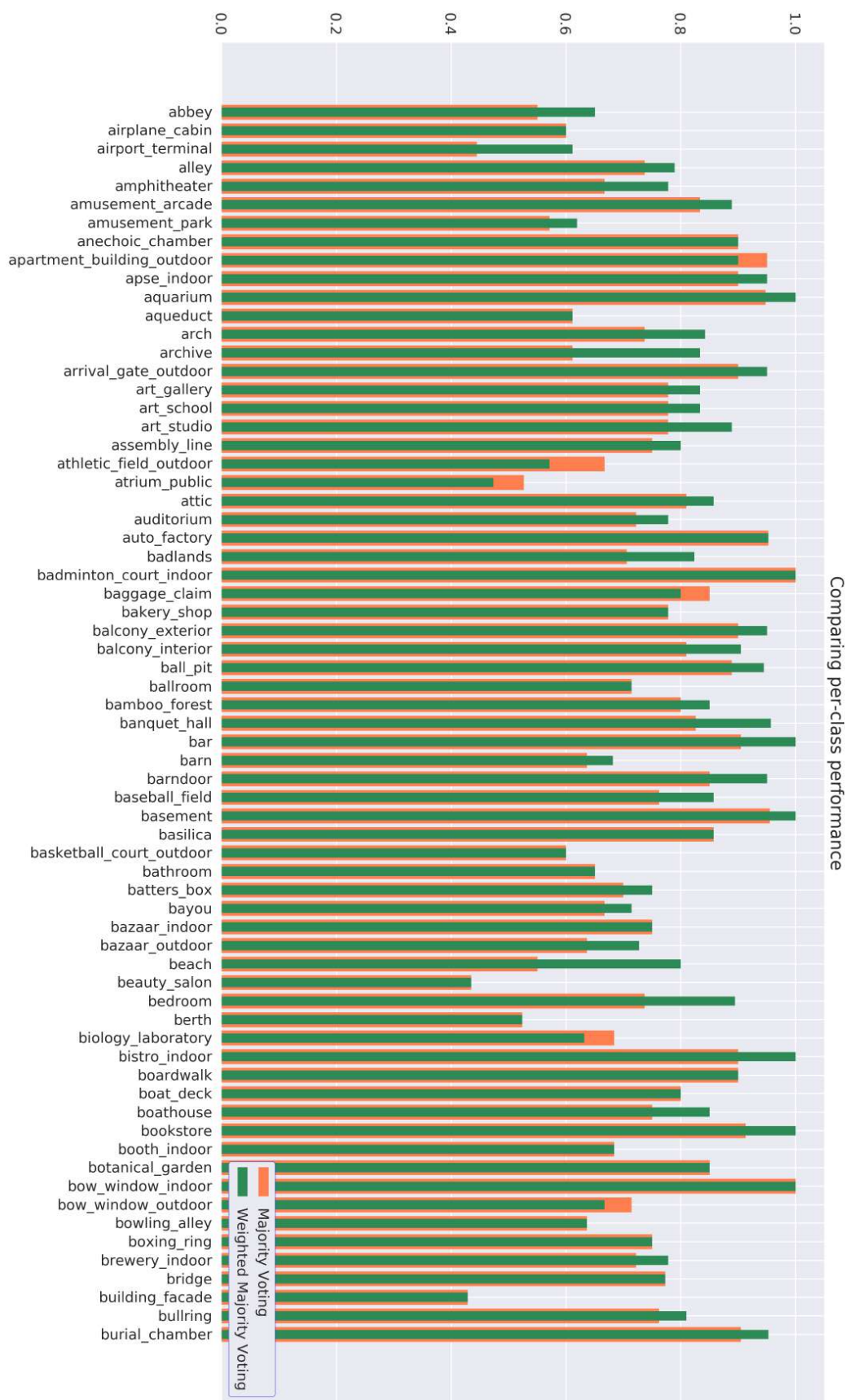


Figure 5.18: Per-class performance on MIT67 with a vanilla majority voting compared with our weighted approach.

Chapter 6

Conclusions

In this thesis, we presented an approach for scene classification through context modeling of indoor scenes. Our proposal was based on the assumption that an RNN-based method is suitable for the problem of indoor scene recognition, since Quattoni and Torralba [2009] affirm that the correlation of object-level information is highly valuable to tackle it, which was later evidenced by several works. Even though there are other approaches on the literature fundamentally based on the same premise, ours achieve the best result amongst RNN-based methods relying solely on object-level features, without adding information from global structures.

We worked on several aspects to propose an RNN-based approach as a solution for the problem of indoor scene recognition. First, the input was modeled as a sequence of object parts, with a slight improvement over works from the literature, which usually establishes constraints of sequence length on the input. We also provided an ablation analysis comparing different LSTM configurations, as well as training procedures (M2O vs M2M), reaching the conclusion that a bidirectional approach is far more superior, while M2M training can be quite relevant by taking into account the relevance of each sequence element. Leading us to construct a weight matrix correlating object categories to each scene, serving as prior knowledge for a weighted majority voting to aggregate outputs from all timesteps. We showed that an intelligent aggregation of outputs can benefit recognition performance.

Finally, our method can improve over state-of-the-art approaches, surpassing their performance by pairing each method with our own in an ensemble of classifiers. The criteria that determines which paired approach should provide the prediction was based on a Random Forest trained on a validation set to separate accurate predictions from misclassifications. We experimented on several statistical measures to find the ones most suiting to our problem.

6.1 Future Work

Although there is literature entirely dedicated to the problem of recognizing indoor scenes, the majority of datasets are not. As a consequence, the literature does not provide their performance only on the indoor classes. Thus, one future work is to run each approach on several mixed datasets, providing their performance on indoor scenes, outdoor scenes and the entire dataset, in order to provide further knowledge on the strengths and weaknesses of each method, allowing the literature on recognizing indoor scenes to tackle the detected weaknesses.

We are specially interested in the benefits of our approach regarding the fact that it provides a prediction of scene category for each object part relative to the remaining context of the image. It has the potential to detect the most discriminative parts of an image, as well the less representative ones, improving even further the performance of recognition. Although there are several works on selecting discriminative regions for image recognition, little has been explored on how a recurrent approach can be beneficial.

Additionally, a recurrent method for scene recognition allows to explore the extensive literature on RNNs, applying it to images. For instance, anomaly detection has been vastly researched for data with temporal dependencies, however there is little evidence on how suitable it is to detect contextual anomalies on images. This could be accomplished with an RNN as a next step predictor, such that given an input feature x_t , corresponding to a ROI from the image, the model would attempt to predict x_{t+1} . After training, the expected prediction error could be modeled, such that when the RNN is presented with an abnormal element, at test time, that deviates from the context of the remaining image, this anomaly would be reflected in the output error. This proposition of tackling anomaly detection is very common on the literature of recurrent models, so it would be very straightforward applying it to images, and it would provide valuable insight on contextual modeling for images through RNNs.

Overall, with this work we expect to instigate future research on recurrent models applied to image classification, since we were able to improve over RNN-based methods mainly by presenting a well suited sequence composition as input to our BiLSTM, as well as combining the recurrent outputs by taking into account their relevance to the input scene.

Bibliography

- Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993--1022.
- Bosch, A., Muñoz, X., and Martí, R. (2007). Which is the best way to organize/classify images by content? *Image and vision computing*, 25(6):778--791.
- Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5--32.
- Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984). Classification and regression trees.
- Byeon, W., Breuel, T. M., Raue, F., and Liwicki, M. (2015). Scene labeling with lstm recurrent neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3547--3555.
- Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.
- Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.
- Csurka, G., Dance, C., Fan, L., Willamowski, J., and Bray, C. (2004). Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, pages 1--2. Prague.
- Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*.
- Espinace, P., Kollar, T., Roy, N., and Soto, A. (2013). Indoor scene recognition by a mobile robot through adaptive object detection. *Robotics and Autonomous Systems*, 61(9):932--947.
- Fei-Fei, L. and Perona, P. (2005). A bayesian hierarchical model for learning natural scene categories. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 524--531. IEEE.
- Felzenszwalb, P. F. and Huttenlocher, D. P. (2004). Efficient graph-based image segmentation. *International journal of computer vision*, 59(2):167--181.

- Gini, C. (1912). Variabilità e mutabilità. *Reprinted in Memorie di metodologica statistica* (Ed. Pizetti E, Salvemini, T). Rome: Libreria Eredi Virgilio Veschi.
- Gong, Y., Wang, L., Guo, R., and Lazebnik, S. (2014). Multi-scale orderless pooling of deep convolutional activation features. In *European conference on computer vision*, pages 392--407. Springer.
- Grzeszick, R. and Fink, G. A. (2016). Zero-shot object prediction using semantic scene knowledge. *arXiv preprint arXiv:1604.07952*.
- He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770--778.
- Herranz, L., Jiang, S., and Li, X. (2016). Scene recognition with cnns: objects, scales and dataset bias. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 571--579.
- Hochreiter, S. (1998). The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107--116.
- Hochreiter, S., Bengio, Y., Frasconi, P., Schmidhuber, J., et al. (2001). Gradient flow in recurrent nets: the difficulty of learning long-term dependencies.
- Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. 9:1735--80.
- Hsieh, T.-J., Hsiao, H.-F., and Yeh, W.-C. (2011). Forecasting stock markets using wavelet transforms and recurrent neural networks: An integrated system based on artificial bee colony algorithm. *Applied soft computing*, 11(2):2510--2525.
- Javed, S. A. and Nelakanti, A. K. (2017). Object-level context modeling for scene classification with context-cnn. *arXiv preprint arXiv:1705.04358*.
- Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*.
- Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097--1105.
- Laranjeira, C. and Nascimento, E. R. (2017). Representing indoor scenes as a sparse composition of feature segments.

- Lazebnik, S., Schmid, C., and Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 2169--2178. IEEE.
- LeCun, Y., Haffner, P., Bottou, L., and Bengio, Y. (1999). Object recognition with gradient-based learning. In *Shape, contour and grouping in computer vision*, pages 319--345. Springer.
- Li, L.-J., Su, H., Lim, Y., and Fei-Fei, L. (2010). Objects as attributes for scene classification. In *European Conference on Computer Vision*, pages 57--69. Springer.
- Li, Z., Lu, W., Sun, Z., and Xing, W. (2018). Improving multi-label classification using scene cues. *Multimedia Tools and Applications*, 77(5):6079--6094.
- Liao, Y., Kodagoda, S., Wang, Y., Shi, L., and Liu, Y. (2016). Understand scene categories by objects: A semantic regularized scene classifier using convolutional neural networks. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 2318--2325. IEEE.
- Liwicki, M., Graves, A., Fernández, S., Bunke, H., and Schmidhuber, J. (2007). A novel approach to on-line handwriting recognition based on bidirectional long short-term memory networks. In *Proceedings of the 9th International Conference on Document Analysis and Recognition, ICDAR 2007*.
- Lloyd, S. (1982). Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129--137.
- Nascimento, G., Laranjeira, C., Braz, V., Lacerda, A., and Nascimento, E. R. (2017). A robust indoor scene recognition method based on sparse representation. In *22nd Iberoamerican Congress on Pattern Recognition. CIARP*, Valparaiso, CL. Springer International Publishing. To appear.
- Normalization, B. (2015). Accelerating deep network training by reducing internal covariate shift. *CoRR.-2015.-Vol. abs/1502.03167.-URL: <http://arxiv.org/abs/1502.03167>*.
- Oliva, A. and Torralba, A. (2001). Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision*, 42(3):145--175.
- Oord, A. v. d., Kalchbrenner, N., and Kavukcuoglu, K. (2016). Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*.
- Pearson, K. (1894). Contributions to the mathematical theory of evolution. *Philosophical Transactions of the Royal Society of London. A*, 185:71--110.
- Pearson, K. (1905). "das fehlergesetz und seine verallgemeinerungen durch fechner und pearson." a rejoinder. *Biometrika*, 4(1-2):169--212.

- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825--2830.
- Pérez-Ortiz, J. A., Calera-Rubio, J., and Forcada, M. L. (2001). Online text prediction with recurrent neural networks. *Neural processing letters*, 14(2):127--140.
- Quattoni, A. and Torralba, A. (2009). Recognizing indoor scenes. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 413--420. IEEE.
- Schuster, M. and Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673--2681.
- Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., and LeCun, Y. (2013). Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*.
- Serrano, N., Savakis, A. E., and Luo, J. (2004). Improved scene classification using efficient low-level features and semantic cues. *Pattern Recognition*, 37(9):1773--1784.
- Sharif Razavian, A., Azizpour, H., Sullivan, J., and Carlsson, S. (2014). Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 806--813.
- Sherstinsky, A. (2018). Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. *arXiv preprint arXiv:1808.03314*.
- Shuai, B., Zuo, Z., Wang, B., and Wang, G. (2016). Dag-recurrent neural networks for scene labeling. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3620--3629.
- Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- Sivic, J. and Zisserman, A. (2003). Video google: A text retrieval approach to object matching in videos. In *null*, page 1470. IEEE.
- Szumner, M. and Picard, R. W. (1998). Indoor-outdoor image classification. In *Proceedings 1998 IEEE International Workshop on Content-Based Access of Image and Video Database*, pages 42--51. IEEE.
- Uijlings, J. R., Van De Sande, K. E., Gevers, T., and Smeulders, A. W. (2013). Selective search for object recognition. *International journal of computer vision*, 104(2):154--171.
- Ulrich, I. and Nourbakhsh, I. (2000). Appearance-based place recognition for topological localization. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, volume 2, pages 1023--1029. Ieee.

- Vailaya, A., Figueiredo, M. A., Jain, A. K., and Zhang, H.-J. (2001). Image classification for content-based indexing. *IEEE transactions on image processing*, 10(1):117-130.
- Vailaya, A., Jain, A., and Zhang, H. J. (1998). On image classification: city vs. landscape. In *Proceedings. IEEE Workshop on Content-Based Access of Image and Video Libraries (Cat. No. 98EX173)*, pages 3--8. IEEE.
- Wang, Y. and Pan, W. (2017). Scene recognition with sequential object context. In *CCF Chinese Conference on Computer Vision*, pages 108--119. Springer.
- Wang, Z., Wang, L., Wang, Y., Zhang, B., and Qiao, Y. (2017). Weakly supervised patchnets: Describing and aggregating local patches for scene recognition. *IEEE Transactions on Image Processing*, 26(4):2028--2041.
- Wöllmer, M., Metallinou, A., Eyben, F., Schuller, B., and Narayanan, S. (2010). Context-sensitive multimodal emotion recognition from speech and facial expression using bidirectional lstm modeling. In *Proc. INTERSPEECH 2010, Makuhari, Japan*, pages 2362--2365.
- Xiao, J., Hays, J., Ehinger, K. A., Oliva, A., and Torralba, A. (2010). Sun database: Large-scale scene recognition from abbey to zoo. In *Computer vision and pattern recognition (CVPR), 2010 IEEE conference on*, pages 3485--3492. IEEE.
- Xie, L., Hong, R., Zhang, B., and Tian, Q. (2015). Image classification and retrieval are one. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, pages 3--10. Acm.
- Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., and Oliva, A. (2014a). Learning deep features for scene recognition using places database. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 27*, pages 487--495. Curran Associates, Inc.
- Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., and Oliva, A. (2014b). Learning deep features for scene recognition using places database. In *Advances in neural information processing systems*, pages 487--495.
- Zuo, Z., Shuai, B., Wang, G., Liu, X., Wang, X., Wang, B., and Chen, Y. (2015). Convolutional recurrent neural networks: Learning spatial dependencies for image representation. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 18--26.
- Zuo, Z., Shuai, B., Wang, G., Liu, X., Wang, X., Wang, B., and Chen, Y. (2016). Learning contextual dependence with convolutional hierarchical recurrent neural networks. *IEEE Transactions on Image Processing*, 25(7):2983--2996.