Enhanced Feedback Enabled Cascaded Classification Models For Flexible Holistic Scene Understanding

Vorgelegt von
David Wilson Romero Guzmán

Von der Fakultät V - Verkehrs- und Maschinensysteme
der Technischen Universität Berlin
zur Erlangung des akademischen Grades
Master of Science
- Informationstechnik im Maschinenwesen -
genehmigte Abschlussarbeit.

Gutachter :	Prof. Dr.-Ing. Olaf Hellwich
	Prof. Dr. Henning Sprekeler
Betreuer :	Dr.-Ing. Ronny Hänsch

Eidesstattliche Versicherung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und eigenhändig sowie ohne unerlaubte fremde Hilfe und ausschließlich unter Verwendung der aufgeführten Quellen und Hilfsmittel angefertigt habe.

Berlin, den 08. Oktober 2018 .....

David Wilson Romero Guzmán

Enhanced Feedback Enabled Cascaded Classification Models For Flexible Holistic Scene Understanding

Abstract: Scene understanding includes several constituent tasks, such as scene categorization, object recognition, depth estimation, etc. Each of these tasks is often considerably hard, and state-of-the-art classifiers already exist for many of them. As these classifiers often operate on the same raw data and provide correlated outputs, it is desirable to dispose of a unified framework able to capture and use such correlation to improve the overall performance, without requiring any changes in the internal working structure of the classifiers. Furthermore, it would be advantageous for the framework to allow for seamlessly inclusion of additional tasks in the model and the possibility to upgrade the existing ones with new emerging techniques. We propose the Enhanced Feedback Enabled Cascaded Classification Models (EFE-CCM), which is able to jointly optimize all the constituent tasks, exclusively requiring a "black-box" interface of the classifiers. We use a two-layer cascade of classifiers, each with a unique instantiation of every task, in which the output of the first layer is included into the second layer as input. As the generated classifier outputs are often correlated, we introduce an inter-layered feature reduction strategy, which eliminates redundant information, reducing the complexity of the second layer classifiers, while keeping its discriminant power. The training strategy involves a feedback step, which allows deeper classifiers to encourage shallower ones to focus on correctly classifying relevant error modes that affect the performance "downstream". We show that our EFE-CCM, more precisely our (LDA)FE-CCM, is able to improve performance in terms of classification accuracy and processing time for the constituent tasks of scene understanding in comparison with the predecessor FE-CCM and CCM models. In our experiments we consider the tasks of scene categorization and object recognition.

Keywords: Scene understanding, simultaneous scene categorization and object recognition, computer vision, machine learning.

Enhanced Feedback Enabled Cascaded Classification Models For Flexible Holistic Scene Understanding

Zusammenfassung: Das Szenen-Verständnis umfasst mehrere konstituierende Aufgaben, wie Szenenkategorisierung, Objekterkennung, Tiefenschätzung, usw. Diese Aufgaben sind oft sehr schwierig zu lösen und für viele von ihnen existieren bereits state-of-the-art Klassifikatoren. Da diese Klassifikatoren oft mit den gleichen Rohdaten arbeiten und korrelierte Ausgaben bereitstellen, ist es wünschenswert, über ein einheitliches Framework zu verfügen, das eine solche Korrelation erfassen und verwenden kann, um die Gesamtleistung zu verbessern, ohne dabei Änderungen in der inneren Arbeitstruktur der Klassifikatoren zu erfordern. Darüber hinaus wäre es vorteilhaft, die nahtlose Aufnahme zusätzlicher Aufgaben und das Aufrüsten der existierenden mit neuen aufkommenden Techniken zu ermöglichen. Wir stellen die "Enhanced Feedback Enabled Cascaded Classification Models" (EFE-CCM) vor, die in der Lage sind, alle kostituierenden Aufgaben gemeinsam zu optimieren, wobei es auschließlich eine "Black-Box"-Schnittstelle der Klassifikatoren benötigt wird. Wir verwenden eine zweischichtige Kaskade von Klassifikatoren, in der die Ausgabe der ersten Ebene als Eingabe in die zweite Ebene eingefügt wird, wobei jeder Schicht aus einer einzigen Instanziierung jeder Aufgabe besteht. Da die generierten Ausgaben der Klassifikatoren häufig korreliert sind, führen wir eine zwischenschichtige Dimensionalitätsreduktiontechnik ein, die redundante Information aus dem Eingangsmerkmalsvektor der zweiten Ebene eliminiert und dadurch die Komplexität der Klassifikatoren in der zweiten Ebene reduziert, während ihre Diskriminanzstärke erhalten bleibt. Die Trainingsstrategie beeinhaltet einen Feedback-Schritt, der es tieferen Klassifikatoren ermöglicht, flacheren zu ermutigen, sich darauf zu konzentrieren, relevante Fehlermodi korrekt zu klassifizieren, die die Klassifikationsleistung "stromabwärts" verschlechtern. Wir zeigen, dass unser EFE-CCM, genauer gesagt unser (LDA)FE-CCM, in der Lage ist, Leistung hinsichtlich der Klassifikationsgenauigkeit und der Bearbeitungszeit im Vergleich zu den Vorgängermodellen FE-CCM und CCM für die konstituierenden Aufgaben des Szenen-Verständisses zu verbessern. In unseren Experimenten betrachten wir die Aufgaben der Szenenkategorisierung und der Objekterkennung.

Keywords: Szenen-Verständnis, gemeinsame Szenenkategorisierung und Objekterkennung, Computer Vision, maschinelles Lernen.

Acknowledgments

I would like to thank Prof. Dr.-Ing Olaf Hellwich and Prof. Dr. Henning Sprekeler for their time in the reviewing process of this work. Furthermore, I would like to express special gratitude to Dr.-Ing. Ronny Hänsch for his excellent assistance in the development of this master dissertation. Besides, I would like to thank my family and friends for the encouragement and support provided during the development of this work and specially my girlfriend whose coffee kept me on the right track.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Throughout history, one of the main goals of computer vision has been that of Holistic Scene Understanding. This requires solving several tasks simultaneously (e.g. object detection, scene classification, depth estimation, ...) each of which attempts to explain some particular aspect of the scene. Due to the increasing interest in Computer Vision, the last decades have seen great progress in tackling each of these constituent tasks, leading to the development of several classifiers with remarkable performance. As these approaches often solve these tasks in an isolated fashion, one very interesting question emerges: Are the constituent tasks of Holistic Scene Understanding correlated? And if so, could these tasks benefit from each other's responses to improve the overall performance? Interestingly, several of these scene aspects are often correlated. As a result, approaches aiming to reason about the constituent tasks of Holistic Scene Understanding often generate correlated outputs too. Consider a scene classifier, whose task is to classify a given image as an indoor or an outdoor scene. Additionally, consider an animal classifier, whose task is to assign animal labels (e.g. dog, cat, spider) to several regions in the image. Imagine, that for a certain region of a particular image, there is an equal affiliation probability of exactly 50% towards both classes "bear" and "dog". Now, imagine that the scene classifier determines an affiliation probability of that same image of 90% towards the class "indoor scene". In this case, one could utilize the contextual information provided by the scene classifier to break the tie in the object recognition task. It is intuitive to set the likelihood of observing a domestic animal in an indoor scene significantly higher than that of observing a wild animal. As a result, under these circumstances, one would prefer to assign the label "dog" to this particular region.

Although the rather intuitive idea of permitting communication between converse tasks in Holistic Scene Understanding dates back to the early days of computer vision, its implementation has proven to be extremely difficult. In fact, only recently have researchers returned back to the difficult task of considering its numerous tasks jointly. In this work, we propose the **Enhanced Feedback Enabled Cascaded Classification Models (EFE-CCM)**, an extension of the Feedback Enabled Cascaded Classification Models [Li 2012] FE-CCM and the Cascaded Classification Models (CCM) [Heitz 2009]. The EFE-CCM defines a generalizable flexible unified framework, which is able to reason jointly about several constituent tasks, allowing them to share information and steer the classifiers towards a joint optimum.

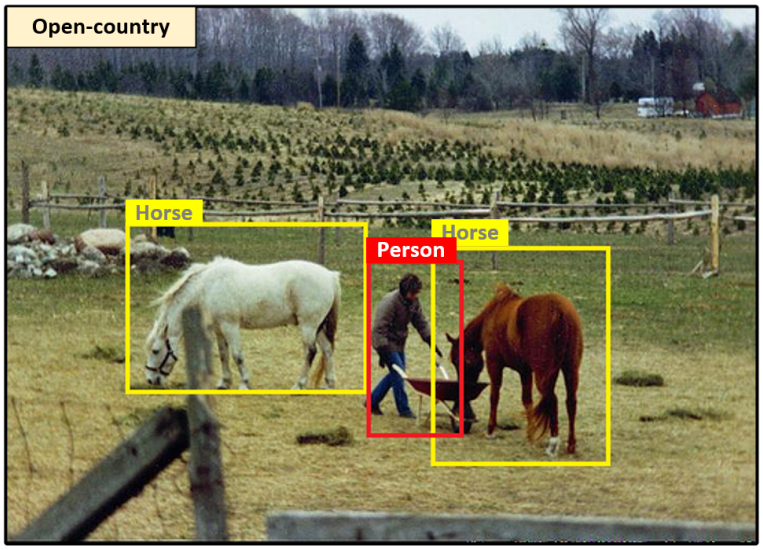


Figure 1.1: Holistic Scene Understanding for simultaneous Scene Categorization and Object Recognition. Given an input image, Holistic Scene Understanding aims to infer labels for all possible scene understanding components. In our study case, we consider the tasks of scene categorization and object recognition but scene understanding can be expanded for simultaneous depth estimation, saliency detection, among others.

Although several joint formulations have been proposed in the last few decades, these strategies are typically restricted to a small subset of the scene understanding tasks [Yao 2012], [Wei 2017]. Moreover, a complex intimate interlacing logic between the constituent tasks is generally required, which hampers any effort to include additional tasks in the model or even to upgrade the existing ones. As a consequence, any intended modification on the framework requires high understanding of the inner structure of the components and the unifying algorithm itself. Heitz et. al. proposed a solution to this issue with their Cascaded Classification Models (CCM) [Heitz 2009]. The CCM is able to conduct joint optimization on its components without imposing any further restriction on their internal operation other than requiring a simple "black box" input/output interface for training and inference. With this formulation, CCM obtains a very high flexibility in comparison with converse Holistic Scene Understanding approaches. In this context, we define flexibility as the capacity of a framework to seamlessly incorporate or upgrade tasks in its formulation. The high flexibility of the CCM allows for the framework to stay "up-to-date" with new emerging techniques.

Unfortunately, the CCM suffers of multiple drawbacks that lead to suboptimality. In short, the CCM is composed of multiple tiers, each equipped with a unique classifier for every constituent task. The tasks are repeatedly instantiated in each layer considering the outputs of the last layer as additional features. Although

these additional features provide contextual information from converse tasks, the CCM trains each tier independently, using the ground-truth labels of each task as the learning objective, what provokes layer-wise isolation. As a result, layers are not able to communicate with each other or transmit information about possible improvement directions other than their actual responses. Furthermore, the learning schema of the CCM is exclusively based on a feed-forward fashion. The lack of a feedback interface makes it impossible for deep layers to encourage shallower ones to focus their efforts on correctly classifying relevant mistakes, while ignoring those that do not hurt "downstream". These properties altogether lead to a suboptimal classification scheme, which exhibits multiple improvement opportunities.

The aforementioned problems were overcome by Li et. al. with their proposed Feedback Enabled Cascaded Classification Model (FE-CCM) [Li 2012]. In counterpart to the CCM, the FE-CCM provides inter-layer communication means and, more importantly, provides a feedback interface. This design scheme enables simultaneous multi-layer optimization, producing a boost in the learning capability of the algorithm and a remarkable performance improvement over its predecessor. Furthermore, Li. et. al. extended the learning schema of the CCM to heterogeneous datasets. In other words, they enabled simultaneous learning on disjoint datasets, each of which contains ground-truth information for a singular task. This extension turns out very useful in practice, as data sets usually contain ground-truth information for a single task. Consequently, this learning scheme extensively widens the applicability of the FE-CCM across multiple research fields and makes it promising for posterior research.

The biggest downside of the FE-CCM is related to its relative high complexity for both learning and inference assignments. While an inference query requires inference on a single classifier for a task in isolation, in an FE-CCM, we need to run inference on $n(L-1)+1$ classifiers, where L describes the number of layers and n the number of constituent tasks. Additionally, the output of the l-th classification layer is concatenated to the original task feature vector, producing an increase in the number of features proportional to the amount of tasks incorporated in the model n, for all classifiers in the $(l+1)$-th layer. This concatenation augments the complexity of the learning and inference tasks on all but the shallowest layer, which simultaneously constraints the applicability of the algorithm for an increasing number of components.

The proposed **Enhanced Feedback Enabled Cascaded Classification Models (EFE-CCM)** aims to strengthen some of the most important deficiencies of the FE-CCM and the CCM:

- As depicted before, the biggest downside of the FE-CCM is related to its inference and learning complexity. To tackle this issue, the EFE-CCM incorporates a systematic procedure to reduce the cardinality of the input features for each constituent task. We achieve this with an inter-layered feature reduction algorithm, which is applied over the feature space of the input feature vector of the subsequent layer. Consequently, our feature reduction does not exclusively affect the original feature vector of every task, but, on the contrary, it reduces the dimensionality of the previous classification layer's response too.

  The most straight-forward implementation of such a feature reduction could be formulated independently for the response of each classifier in the previous layer, whose results are subsequently concatenated to form the input vector of the following layer. However, as stated previously, the tasks involved in holistic scene understanding are often correlated. This suggests that the responses provided by each task classifier are likely to be correlated as well. Based on this, we argue that the appliance of a feature reduction technique on the joint response of the classification layer can deliver superior results.

  The final goal of our feature reduction approach is to merge several classifier responses of a layer into a compact feature vector containing the exact same descriptor power and so, reduce the complexity of the cascaded model, while beholding (most of) its discriminative capacity.

Although the main focus of this work is directed towards joint object recognition and scene classification, we provide generalized formulations that allow for seamless inclusion of additional tasks.

In extensive experiments, we evaluate the performance of the EFE-CCM over the FE-CCM, the CCM and the utilized baselines in terms of its classification accuracy and processing time. Since neither information on their processing time nor source code for the CCM nor the FE-CCM has been provided by the authors, we reimplement these frameworks based on their available descriptions in [Li 2012] and [Heitz 2009], respectively[1]. Subsequently, we confront the obtained improvements and degradations and analyze their corresponding causes.

The rest of the document is organized as follows: a discussion on theoretical background and related works is carried out in Chapter 2. Subsequently, in-depth descriptions of the CCM, the FE-CCM and our proposed EFE-CCM are provided in Chapter 3. Our implementation details as well as a brief description of our baselines are given in Chapter 4. Our experimental settings and obtained results are discussed Chapter 5. To finalize, the Section 6 summarizes conclusions, commentaries and possible future work directions.

[1] Our implementation is available online at http://github.com/RomeroGuDw/master-thesis

Chapter 2

Overview on Scene Understanding

Contents

In this section we define holistic scene understanding, determine some of its challenges and describe how CCM-structures intent to confront them. Subsequently we provide a thorough review on works related to holistic scene understanding and compare them with our proposed framework.

2.1 Holistic Scene Understanding

When we analyze an image of a scene, we are often interested in answering several different questions: Which type of scene is it? What objects are there? How many objects of each class are shown? Where are they located? How far are things in the scene? and so on. These are just a few examples of the questions that emerge in scene understanding. In the past, many authors have approached each of these questions independently, where the goal is to infer an appropriate label $Y_i \in \boldsymbol{S_i}$ for the i-th subtask. For instance, in scene categorization we are interested in obtaining a label $Y_{SC} \in \{1, 2, ..., K\}$, that correctly assigns an image to a group of K disjoint scene types. Conversely, in depth estimation we want to assign a label $Y_{DE} \in \Re_+$ to each pixel in an image so that it matches with its corresponding depth. Under a scene understanding schema with n subtasks, the desired output is a vector $\boldsymbol{Y}$, such that

$$\boldsymbol{Y} = \{Y_1, ..., Yn\} \in \boldsymbol{S_1} \times \boldsymbol{...} \times \boldsymbol{S_n}$$

corresponds to the optimal MAP assignment for all the constituent tasks. As depicted before, n can variate drastically depending on the questions asked. In

order to achieve scene understanding for any desired combination of tasks, a flexible algorithm that does not impose any constraints on its components is required.

Holistic scene understanding as well as its components are complex tasks that have been under the loupe for many years. Their complexity is closely related to the large quantity of elements that need to be considered when modeling a solution approach and the intrinsic connection between them. Scene understanding yields to a mixture of low, mid and high-level features that describe scenes at multiple levels, starting from pixel-wise texture distribution up to diverse relationships within objects in the scene. Additionally, scene modelling involves semantic abstract cues, such as object relative locations and spatial appearance tendencies within an image, which need to be jointly analyzed to get an in-depth understanding of an scene. While the recollection of such information is trivial for human beings, it is still difficult to obtain with artificial systems. The CCM and its posterior extensions intent to obtain a large amount of information from the entire range of features (low, mid, high) and the present abstract cues in a hierarchical manner (Figure 2.1). The constituent tasks of a CCM generate labels for a specific topic based on the combination of low and mid-level features. During this process the classifiers capture relevant relationships between the features and utilize them to generate an appropriate mapping from the features to higher label features and clues (label inferences). Subsequently, CCM utilizes the generated high-level features to capture relationships between the labels and reinforce the beliefs of each constituent block.

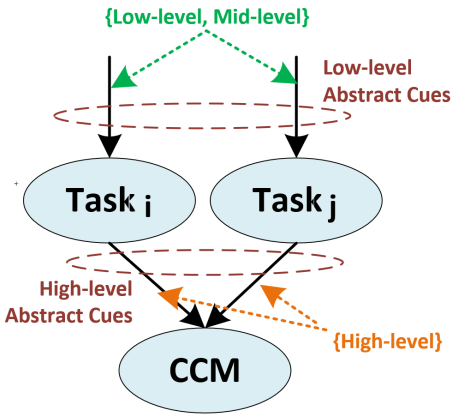


Figure 2.1: Hierarchical inclusion of multi-level feature information in the Cascaded Classification Model. While the constituent tasks of the CCM reason about low- and mid-level cues, the CCM combines the inferred high-level responses to generate additional information, with which the classifiers can be driven towards a more descriptive model.

2.2 Related Work

In this section we provide a thorough review on related works that aim to combine classification tasks as well as multi-source information into a unique framework and on multiple dimensionality reduction techniques. Initially, we introduce multiple classification techniques based on cascaded classifiers and review initialization techniques utilized for these models. Subsequently, we analyze several developed techniques for sensor fusion, structured modeling and context modeling to converge to a review of several works related to joint classification both in holistic scene understanding and in other machine learning fields. To finalize, we review several dimensionality reduction strategies.

2.2.1 Cascaded Classifiers

The idea of cascading layers of classifiers to reinforce the task goal was first introduced with neural networks as a composite of multilayer perceptrons. In this scheme, the concatenated output of a perceptron layer is passed on to each perceptron of the next layer as input [Hansen 1990], [Rowley 1998]. Posterior extensions on neural networks have variated the way the information is passed from one layer into the next, e.g. Convolutional Neural Networks [Krizhevsky 2012], Recurrent Neural Networks [Graves 2009], Modular Neural Networks [El-Bakry 2001], among others. Convolutional Neural Networks (CNN) constraint the information passing schema of conventional neural networks to a region around the current perceptron. This scheme is utilized to reduce the connectivity of the layers and reduce the complexity of the tasks and the number of required samples for proper learning. One important problem of neural networks is the inability to gain insight into their internal operation and learned concepts, which make it difficult to be used for structured tasks or to add extra information to individual classifiers within the network. This situation is hardly aggravated when the number of hidden layers increase, e.g. Deep Learning [Collobert 2008]. Due to this problem, researchers implemented the idea of cascading to many other kinds of problems pursuing structures, whose inner structure was easier to understand and modify. One of the most spread techniques in literature is that of Boosting [Freund 1997], where many weak classifiers are combined using arithmetical operations on their outputs to obtain a more accurate classifier. Boosting has been applied for several tasks such as object recognition [Zhang 2006] and face detection [Huang 2005a]. In order to refine the boosting framework, Fink and Perona [Fink 2004] incorporated contextual information exploiting local dependencies between objects. Subsequently, Torralba et. al. [Torralba 2005] introduced Boosted Random Fields to model object dependencies which used boosting to learn the graph structure and the local evidence of a conditional random field. All these works mainly consider the interaction between labels of the same type in an exclusive feed-forward manner. Conversely, CMM aims to capturing contextual interaction between labels of different natures. Furthermore, due to their feedback interface, FE-CCM and EFE-CCM allow refinement not just of the contextual interactions but

of the individual classifiers as well, providing more relevant and helpful information for subsequent classification layers.

While efficient back-propagation methods have been commonly used in the learning of multilayer networks [LeCun 1998], it is not easy to implement these methods in structures such as CCM [Heitz 2009] or FE-CCM [Li 2012], where each node is a complex classifier. Interestingly, some structured learning schemas were recently proposed, which combine deep-learning strategies with conditional random fields [Yu 2010], [Paisitkriangkrai 2015]. We argue that such methods could be of interest for subsequent holistic scene understanding strategies and will be analyzed in future work. A very important consideration when learning with hidden variables is their initialization. Optimization methods for learning usually lean on hill-climbing-like strategies, which are prone to local optima stagnation. In order to avoid such situations, an intelligent parameter initialization is crucial to obtain good results. Common strategies for parameter initialization are swarm-like random initialization [Kennedy 2011], prior knowledge-matching initialization [Heckerman 1995] and unsupervised learning rounds for initialization [Hinton 2006]. Hinton et. al. [Hinton 2006] utilize unsupervised learning to obtain an initial configuration of the model parameters. This strategy provides a good initialization, with which their multilayered architecture does not suffer from local optima during optimization. CCMs can be described as a multilayered architecture with complex node internal structures. In such an architecture, a good initialization can be established by initializing each node independently, as described by Li et. al. in [Li 2012]. With this initialization, our training procedure learns parameters for all the nodes, which consistently improve the overall classification performance of the model.

2.2.2 Sensor Fusion

Other interesting group of works are those approaching sensor fusion. Sensor fusion relates to the idea of fusing diverse classifiers $C_1, ..., C_n$ that work over heterogeneous information to generate labels from an unique shared set of classes $\boldsymbol{S}$. Due to variated working modalities and different information sources, sensor fusion aims to capture additional information from each classifier to reinforce each other beliefs and obtain better performing accuracy. This strategy has been extensively used in the field of biometrics [Dieckmann 1997], [Ross 2003], [Faundez-Zanuy 2005] and health assessment [Dong 2007], [Yi 2014]. Dieckmann et. al. [Dieckmann 1997] combine voice recognition and face recognition to improve person recognition. Likewise, Dong et. al. [Dong 2007] combine several patient information from multi-sensor equipment to provide more accurate diagnosis and prognosis, such as the likelihood of a patient of having a disease or his propensity to acquire it. Although CCMs could be used with minor modifications for such applications (See Chapter 3), in the scenario of holistic scene understanding we consider multiple tasks, where each classifier approaches a different problem, i.e. generates different labels, based on the same (or related) information.

2.2.3 Structured Models

While Sensor Fusion strategies combine classifiers to predict the same labels, there is a large group of works that design structured models to predict heterogeneous labels. These models are usually related with Probabilistic Graphical Models (PGMs) in diverse instantiations (e.g. MRF, CRF, BN). We refer the reader to [Koller 2009] for a nice and complete overview of PGM architectures, representation methods, inference strategies and learning schemes. In short, a Probabilistic Graphical Model (PGM) is a general-purpose framework for constructing and using probabilistic models of complex systems, characterized by the presence of multiple interrelated aspects, many of which relate to the reasoning task. The task of reasoning probabilistically about the values of one or more of the variables, subject to possible observations over some other variables, require the construction of a joint distribution over the space of possible assignments to some set of random variables $\boldsymbol{X}$. Probabilistic graphical models allow us to answer a broad range of interesting queries over $\boldsymbol{X}$. For example, if we make an observation over the variable X_i of taking the value x_i, we could ask what the probability distribution over values of another variable X_j in the resulting posterior distribution is. Consider that for a particular application all variables X_i in the model can take 64 different values. When this consideration is merged with the fact that in practice, probabilistic inference problems are usually constituted by dozens or even hundreds of relevant attributes, e.g. label of each pixel in an image, the inference problem might become intractable. PGMs provide mechanisms to exploit structure in complex distributions to describe them compactly in a practical way to construct and utilize them effectively. Another very valuable characteristic of PGMs is their ability to seamlessly aggregate prior information to the model, e.g. from field experts. This ability heavily contrasts with many other machine learning structures, where prior information is often burdensome to include. Due to these favorable characteristics of PGMs, PGMs have found application throughout several fields of machine learning, such as 3D Robot Mapping [Thrun 2004], collective Clustering for body-part recognition [Anguelov 2005], route Alignment of Helicopter trajectories [Abbeel 2007] and speech Recognition [Sung 2009]. In the field of computer vision, PGMs have been used for roughly every possible task in the field, variating from denoising [Malfait 1997], [Buades 2005], segmentation [Bouman 1994], [Deng 2005], object recognition [Cohen 1991], [Quattoni 2005] to scene classification [Wang 2007] and action recognition [Wang 2009], [Zhang 2010].

There has been a large development in structured learning on handling latent variables as well, e.g. hidden conditional random field [Liu 2015],[Yu 2010], [Paisitkriangkrai 2015], latent structured SVM [Zhu 2010], [Wu 2013], [Durand 2015]. The inclusion of latent variables in a model allow for the modelling and aggregation of additional concepts, whose existence we know but cannot measure. We relate the reader to [Rabiner 1986] for a nice introduction in hidden variables and their usage in Probabilistic Graphical Models. One advan-

tage of using latent variables is that they serve to reduce the dimensionality of the data. For example, a large number of observable variables (e.g. sensor responses) can be aggregated to a single underlying concept (e.g. current state) to obtain a better and easier representation of the data. This kind of structures are broadly used to model temporal data sequences in a probabilistic fashion. In [Thrun 2004], Thrun et. al. utilize a Hidden Markov Chain to model the current position of the robot (latent variable) from noisy sensor measurements (observable variables) and the later n time steps. By utilizing hidden variables to model the actual state of the robot, the transition probabilistic model $P(X_{t+1}|X_t, X_{t-1}, ..., X_{t-(n-1)})$ is substantially reduced from a joint probability distribution over $n * m$ variables to a probability distribution over n variables, where m corresponds to the total number of considered sensors in the model and n relates to the state of the robot in the last n time steps. The usage of Latent Variables in the FE-CCM follows a slightly different purpose. While in the later example hidden variables are used to provide a simplified yet highly descriptive model reducing the dimensionality of the data, Li et. al. [Li 2012] utilize hidden variables to increase the flexibility of the model, allowing the ground-truth of the hidden layers to disagree with the actual ground-truth labels of the task. This methodology allows for the hidden layers to focus learning towards error modes that improve the accuracy in subsequent layers. The usage mode of hidden variables to increase the flexibility of the model is closely related with the learning strategy used in models such as deep learning. We refer the reader to [Bengio 2007] for a complete overview of deep learning architectures and to [Caruana 1998] for an overview in multi-task learning with shared representations.

2.2.4 Context Modeling

There is a large body of works that leverage contextual information to aid a specific task. In the computer vision field, a large number of context sources has been explored, ranging from the global scene layout, to interactions between scenes, objects, regions, attributes, segments and local features. In the early work of Galleguillos and Belongie [Galleguillos 2010], the context refers to three main types of contextual information that can be exploited in computer vision: (1) the **semantic context**, which refers to the likelihood of finding objects in some scene types but not in others. These relationships are commonly modelled as co-occurrence probabilities of multiple objects with or in certain scene types; (2) **spatial context**, which refers to the likelihood of finding objects in certain positions relative to other objects or the typical scene layout; and (3) the **scale context**, which exploits the scale relationships within objects in the scene. Li et. al. [Li 2012] utilize object classifier responses to capture relationships within objects and scene types relative to their position, relative scale and co-occurrences to aid scene classification. Hoeim et al. [Hoiem 2008b] utilize 3D scene-layout configurations to provide priors on potential object locations. Suddert et. al. [Sudderth 2006] utilize object recognition to improve 3D-structure estimation. Cardinal et. al. [Gonfaus 2010] incorporate the output of an object detector as image evidence to improve segmentation, while

Lempitsky et. al. [Lempitsky 2009] utilize object detector responses as bounding box priors for object segmentation. Later works have included additional contextual meta-information to aid classification and detection tasks. For example, in the object recognition field, Zhao and Zhu [Zhao 2013] defined objects by integrating their functionality, geometry and appearance information. Choi et. al. [Choi 2010] leverage contextual information in a hierarchical manner. Object hierarchy refers to an specialized research line of object co-occurrences under the assumption that objects are related with a semantic hierarchy, e.g. transport vehicles $\longrightarrow$ {bicycle, car, motorbike, ...}. With an increased number of object categories, object relationships are naturally exhibited as a hierarchical structure learned from hundreds or thousands of object categories in a high-level semantic structure, which is automatically obtained from data [Deng 2011] or with ad-hoc modeling [Zweig 2007]. Due to the fact, that many practical real-life problems are related to human interaction with objects, the problem of reliable human detection has been a topic of high relevance in research over the last decades. Gupta et. al. extended object localization and recognition to human-centric scene understanding by inferring human and 3D-scene interactions [Gupta 2011] and combined spatial and functional constraints between humans and objects to aid accurate recognition of actions and of the involved objects [Gupta 2009]. Prest et. al. [Prest 2012] inferred spatial information of objects by modeling relationships between human bodies and objects, while Park et. al. [Park 2010] utilized ground plane estimation as support information to improve pedestrian detection.

Many works model context to capture local interactions between neighboring regions and low-level features as well. Rabinovich and Belongie [Rabinovich 2009] proposed a classification of contextual models consisting of models with contextual inference, that based on the statistical summary of the scene, referred to as Scene Based Context Models (SBC) and relationships among objects in the image, referred to as Object Based Context (OBC). To incorporate scene-level information, Torralba and Oliva analyzed and included the statistics of low-level features in the scene to prime object detection [Torralba 2003] and depth estimation [Torralba 2002]. A common characteristic to all the aforementioned techniques is the continuous improvement in comparison with their baselines without considering contextual information in the model. However, most of these methods utilize the support information and the support tasks exclusively as a source of additional information, without considering any improvement strategy on the support sources. This fact makes most of them inadequate for Holistic Scene Understanding as a formulation of a joint optimization problem. Moreover, these methods are usually not implementable to cases where we only dispose of "black-box" classifiers for the individual tasks, since complex interlacing logic between context sources and the classification framework are often required.

2.2.5 Holistic Scene Understanding and Joint Optimization Formulations

Although the idea of tackling multiple correlated tasks in a joint manner might be very intuitive for several machine learning problems, specific formulations of the task are often cumbersome. This can be attributed to the complexity of the underlying variable interaction structure and the high number of required hyperparameters during modeling, e.g. the type of potentials required and their mathematical formulation itself. Towards this complicated task, several researchers have created a large body of models aiming to utilize information from multiple tasks to reinforce their independent beliefs. Due to the rich description capability of structured models and their ability to model complex variable relation structures in a relative low-dimensional space, several of the developed approaches towards this task are based on multiple variations of structured models (See Chapter 2.2.3), mainly on instantiations of probabilistic graphical models (PGMs). Often, these approaches integrate hidden variables to generate a more adequate model of the underlying structures, without heavily increasing the model complexity. In the field of natural language processing, Sutton et. al. [Sutton 2005] combined a parsing model with a semantic role labeling model into a unified probabilistic framework that solved both simultaneously. Subsequently, Collober et. al. [Collobert 2008] proposed a unified architecture for natural language processing that produced part-of-speech tags, chunks, named entity tags, semantic roles, semantically similar words and the likelihood of a sentence to simultaneously make sense both grammatically and semantically using a unified model.

The goal of holistic scene understanding dates back to the early days of computer vision. However, due to its complexity, only recently have researchers returned back to the task of considering its constituent tasks in a joint manner. Roughly all the approaches towards holistic scene understanding tackle exclusively a subset of all the possible inference tasks that could be answered from a single image (Scene Classification, Object Recognition, Depth Estimation, Action Recognition, ...). Therefore, the term Holistic Scene Understanding has been used to name the joint solution of a subset of these tasks as well, e.g. Object Recognition and Object Segmentation. In this section, we adopt this definition and provide a review on works that aim for the simultaneous solution of two or more inference tasks in the computer vision field. A large body of works tackle two inference tasks in a joint fashion. Within these works, the most occurrent combination is that of joint object recognition and segmentation. Towards this task, authors have developed large MRF-based probabilistic models [Kumar 2005], region-based models [Gould 2009], CRFs [Winn 2006] and have combined texture, shape, context and location features into a theme and region based multi-feature fusion model [Jing-Xia 2015]. Sun and Savarese [Sun 2011] utilize an articulated part-based probabilistic model for joint object detection and pose estimation. Hoeim et. al. [Hoiem 2008b] proposed an innovative ad-hoc system to combine boundary detection and surface labeling

by sharing low-level information between the classifiers. While Suddert et. al. [Sudderth 2005] related scenes, objects and object parts in a generative model for joint object and scene recognition, Wojek et. al. [Wojek 2008] combined them into a joint CRF and Bappy and Roy-Chowdhury [Bappy 2016] interconnected multiple CNNs towards the same task. Wang and Ji [Wang 2013] proposed a unified probabilistic approach for attribute prediction and object recognition. Some other works introduce additional tasks into the joint task optimization scheme. Hoiem et. al. [Hoiem 2008a] propose a modular innovative system for integrating object recognition, surface orientation estimation and occlusion boundary detection into a single framework. Conversely, Li et. al. [Li 2009] combined image classification, annotation and segmentation in a hierarchical graphical model. Souiai et. al. [Souiai 2013] combine object segmentation, object recognition and scene labeling into a single convex optimization problem by incorporating an ad-hoc hierarchical relationship map between objects and scene types.

Some other works aim to obtain a more integral version of holistic scene understanding by incorporating several correlated tasks into a unique framework. Wang et. al. [Wang 2015] combine 3D object detection, pose estimation, semantic segmentation as well as depth reconstruction into a single CRF, integrating semantics, geometry as well as geographic information coded as 3D scene priors from freely available maps. Yao et. al. [Yao 2012] proposed a large CRF-based probabilistic model to approach holistic scene understanding. In their paper, they propose a framework that simultaneously reasons about regions, location, class and spatial extent of objects, as well as the type of scene. Lately, Wei et. al. [Wei 2017] proposed a stochastic hierarchical spatial-temporal graph to tackle the problem of 4D Holistic Scene Understanding (i.e. holistic scene understanding considering temporal data). In their formulation, they tackle the tasks of event recognition, sequence segmentation and object localization simultaneously for joint inference.

Common to all the aforementioned approaches is a continuous performance improvement for the constituent tasks when tackled in a joint manner in comparison with their solution in isolation. Unfortunately, all these approaches also share the characteristic of including complex interlacing relationships within the tasks, which makes it extremely difficult to aggregate additional tasks or even to modify the existing ones. In order to do so, in-depth understanding of the inner workings of each component and the joint framework itself is indispensable. In contrast, our Enhanced Feedback-Enabled Classification Model aims for flexible holistic scene understanding without imposing any other constraints to the constituent tasks more than possessing accessible inference and learning functions, such as the Cascaded Classification Model of Heitz. et. al. [Heitz 2009].

2.2.6 Dimensionality and Feature Reduction Strategies

During the last decades, the data-collection industry has undergone a tremendous revolution due to the accelerated development of sensoric instrumentation and database infrastructures [Sorzano 2014]. This phenomenon has caused the generation of enormous amounts of data, which simultaneously has unleashed an exponential growth of many databases and the creation of several benchmarks in the machine learning field. However, the effect of these new sensoric solutions come two-fold. On one side, datasets including a vast amount of samples, such as ImageNet [Deng 2009], LabelMe [Russell 2008] and MS-COCO [Lin 2014], have improved the performance of several machine learning frameworks and made feasible the training of complexer machine learning structures such as Deep Learning and converse derivates from Artificial Neural Networks (CCN, RNN, ...), which would otherwise perform poorly. On the other hand, however, these modern devices usually generate extremely high dimensional information for each sample. As a result, a single sample $Y = y_i$ generally depends of a high dimensional feature vector $\boldsymbol{X} \in \Re^N$. This phenomenon is referred to as the *curse of dimensionality* or the *small-n-large-p effect*. For example, in the biomedical domain, standard microarray datasets usually are composed of thousands of variables (genes) in dozens of samples [Mwangi 2014]. In the computer vision field, hyperspectral cameras allow for better discrimination between several subtle objects and materials by generating hyperspectral images composed of hundreds of contiguous spectral bands, where each band corresponds to a single channel $N \times M$ image [Zhang 2014]. This situation is not restricted to the aforementioned fields. In fact, many other scientific fields such as time series analysis, internet search engines, automatic text and speech analysis, have seen an explosion in the number of variables measured in a single experiment. An additional effect to be considered when increasing the number of features disproportionately in comparison to the number of available training samples is the *Hughes effect* [Hughes 1968]. This effect refers to an increasement of classification accuracy when adding additional information (features) to the model, which is replaced by a subsequent accuracy diminishment after surpassing a given threshold. For instance, in hyperspectral imaging, increasing the number of spectral bands in the model initially produces a boost in the classification accuracy which is subsequently replaced by an accuracy diminishment as a function of the number of spectral bands [Imani 2015]. As a result, statistical and machine reasoning methods face a formidable problem when dealing with situations containing small ratios between number of training samples and the number of features in the data. In order to cope with this problem, the number of input variables is reduced before entering the learning procedure. In general, this process can be tackled in two different ways: (1) By keeping the most relevant variables from the original dataset (*feature selection*) or (2) by exploiting the redundancy of the input data and finding a smaller set of new variables, each produced as a combination of the input variables, which basically contain the same information as the input variables (*dimensionality reduction*). Although there is a large body of work for each of the aforementioned categories [Yang 1997], [Guyon 2003],

[Peng 2005], [Saeys 2007], [Phinyomark 2012], we consider dimensionality reduction techniques to be technically more appropriated as those within the feature selection category. While the later ones aim to behold all the information present in the dataset, feature selection techniques are known for a loss of descriptive power proportionally to the number of eliminated variables, when the eliminated variables are not linear-dependent with any other remaining variable [Guyon 2003].
Feature extraction can be performed in three general ways: unsupervised, supervised and semi-supervised. While unsupervised methods exclusively consider the feature descriptor vectors, supervised methods include the corresponding labels in the feature reduction problem as well. Semi-supervised approaches utilize both labeled and unlabeled samples to find the most adequate feature space conversion. Within the unsupervised methods, the most popular method is the Principal Component Analysis (PCA). PCA was initially introduced over a century ago by Karl Pearson [Pearson 1901] as an analogue of the principal axis theorem in mechanics. PCA constructs relevant features from a linear transformation over the original (possibly correlated) features into a smaller set of orthogonal uncorrelated variables, also known as principal components. Since this is an unsupervised method, it does not consider the actual label-wise distribution of the samples over the feature space. Instead, it works under the assumption that the spreading of the label-wise distribution is maximal along the axis with maximal variance in the feature space. Under this design scheme, PCA aims to obtain a smaller set of orthogonal features that optimally explain the variance in the dataset, subject to a maximal allowed number of features in the new feature space. Although PCA has been widely utilized in every dimensionality reduction problem, it possesses some drawbacks, which under circumstances can lead to poor results. For instance, when the main underlying assumption of PCA is not met, i.e. when the label-wise distribution of the samples does not coincide with the highest-variance directions in the dataset. Additionally, PCA just considers linear dependency between variables, leaving open a possible higher reduction rate when utilizing non-linear relationships between variables. These drawbacks have impulsed researchers to extend PCA to cope better with such situations. Within these large body of work, we can find many non-linear variants of PCA such as Principal curves [Hastie 1989], surfaces [LeBlanc 1994] and manifolds [Zhang 2004], Kernel PCA [Schölkopf 1997] and Robust PCA [Netrapalli 2014]. Converse methods such as Independent Component Analysis (ICA) [Lee 1998] are able to handle situations, in which the label-wise distribution does not follow the maximum variance directions in the dataset but rather can be considered as a combination of underlying independent information components. ICA aims to separate the dataset into a set of independent and relevant features under the assumption that the dataset is composed of multiple statistically independent sources, combined by an unknown but linear mixing process. For a nice and thorough review of unsupervised dimensionality reduction techniques and their mathematical foundations, we refer the reader to [Sorzano 2014].

Although an extensive body of work has reported remarkable results obtained from unsupervised techniques [Liu 2005], [Maas 2013], [Stowell 2014]; it is intuitive to infer that supervised techniques should perform better [Imani 2015], as they consider the actual label-wise distribution of the samples in the data set. However, several supervised techniques suffer from very strong restrictions, which severely constrain their usage for a large number of applications. The most popular supervised feature extraction method is the Linear Discriminant Analysis (LDA) [Izenman 2013]. Unlike PCA, where no difference is taken into account for any class, LDA explicitly attempts to model the difference between classes in the data. LDA maximizes the between-class scatter matrix (S_b) while simultaneously minimizing the within-class scatter matrix (S_w). In order to simplify the solution approach, LDA assumes multivariate normality, homoscedasticity (i.e. identical covariance for all classes) and sampling independence between samples. It has been suggested that LDA is relatively robust to slight violations of these assumptions [Lachenbruch 1979] or under the usage of dichotomous variables (where multivariate normality is often violated) [Klecka 1980]. However, it is strongly recommended to consider further alternatives when experimenting hard violations of the assumptions. Another very important and limiting restriction of LDA is, that LDA can extract a maximum of $c-1$ features from a c-class classification problem, which can lead to significant loss of information or restrict its applicability if the classification error estimates establish that more features are required. Furthermore, LDA fails in cases where the discriminatory information is not attached to the mean but rather in the variance of the data[1]. As well as by PCA, a great deal of researchers has developed extensions to cope with these limitations. For instance, the Generalized Discriminant Analysis (GDA) is the non-linear kernel version of LDA [Baudat 2000]. This version, however, is still restrained to a maximum extraction of $c-1$ features. The Nonparametric Weighted Feature Extraction (NWFE) uses the weighted mean for calculation of nonparametric scatter matrices, which allow for extraction of more than $c-1$ features in a supervised manner [Kuo 2004]. Likewise, the Kernel Nonparametric Weighted Feature Extraction (KNWFE) [Kuo 2009] is the non-linear kernel-extended version of NWFE.

One interesting and relevant field in which feature reduction is an indispensable step of the classification task is that of hyperspectral imaging. In short, hyperspectral imaging gathers and processes information across a large section of the electromagnetic spectrum into several wavelength bands. Whereas the human eye detects visible light mostly in three bands (long wavelengths: red, medium wavelengths: green and short wavelengths: blue), hyperspectral images are typically composed of hundreds of spectral bands. As a result, hyperspectral images are able to distinguish many subtle objects and materials. Hyperspectral images can be interpreted as the superposition of several monochromatic images generated at different wavelengths,

[1] This fact already suggests that a combination of LDA and PCA could be beneficious for the feature reduction task.

i.e. a 3D-image, where the coordinates $\{X, Y\}$ determine the spatial distribution of the elements in the image, while $Z \in \{1, ..., N_b\}$ determines the index of the z-th wavelength band. In hyperspectral imaging, a large body of work both for unsupervised and supervised feature reduction techniques can be found [Fauvel 2013], [Jia 2013]. One very handy endorsement that emerges in hyperspectral imaging is the so-called spatial-feature extraction, which considers the spatial distribution of the objects and labels in the space as well [Fauvel 2013]. As stated in Chapter 3.4, we are able to formulate the feature reduction problem approached in this work as a combination of two smaller ones, one of which corresponds to a multispectral feature reduction problem, allowing us to incorporate spatial information into the feature reduction formulation adequately. For a nice review on spectral-spatial feature reduction techniques, we refer the reader to [Fauvel 2013].

Chapter 3

Cascaded Classification Models

Contents

In this section we present a comprehensive description of the Cascaded Classification Models. The chapter is divided in 4 sections. In the first part we cover general topics for all the CCM instantiations. We define important concepts and introduce the formal notation used for the formal description of the frameworks. The following parts cover the Cascade Classification Model of Heinz et. al. (CCM) [Heitz 2009], followed by the Feedback Enabled Cascaded Classification Model of Li et. al. (FE-CCM) [Li 2012] to converge to our proposed Enhanced Feedback Enabled Cascaded Classification Model (EFE-CCM). Each model description encompasses the model assumptions, considerations as well as relevant observations for our work.

3.1 General Definition and Considerations

In General, a Cascaded Classification Model (CCM) is a multi-objective classification framework composed of multiple tiers, each of which contains a unique classifier instance of each classification task in the model. These layers are stacked on top of each other, giving name to the Cascaded Classification Model. While going downstream, the outputs of a layer are concatenated to the original features of each task

in the following tier, utilizing them as additional features. These concatenations enable communication and information interchange between the tasks, simultaneously creating an interface to exploit correlations between the tasks to jointly improve classification performance of the components. The model implicitly requires for the included classification frameworks to seamlessly accept additional features. This requirement might restrict the usage of particular classification strategies. However, the model does not impose the classifiers of the same task to share the same mathematical formulation across layers, which still enables the usage of such classification strategies in the first layer of the CCM (for those instantiations that do not require feature concatenation). An additional important consideration to keep in mind while formulating the CCM is related to the generated classifier responses. As the purpose of the classifier outputs in earlier layers is to give insight to the subsequent classifier layers into the beliefs distribution for a given instance, a classifier that generates probabilistic responses is preferred. For classifiers that generate labels instead of confidence estimates, we convert the original classifier output to the corresponding odds-ratio. To summarize, it is opportune for the constituent classifiers to: (1) provide a mechanism to include additional (auxiliary) features from other models and (2) to generate a confidence estimate rather than a label.
In the field of scene understanding, a great deal of independent research into each of the vision subtasks has led to excellent classifiers. These classifiers are usually trained on disjoint data sets with their own specialized data structures, features, inference and training algorithms. Due to the high variance between solution strategies, it is convenient to treat each classifier as a "black-box" to facilitate their compatibility, while ignoring their internal working. A formal description of the term "black-box classifier" is given below:

Black-box Classifier: As it name suggests, a black-box classifier denotes a classifier whose learning and inference algorithms are available for use but whose internal operation workings are not known. We assume that, given a training dataset $\boldsymbol{X} = \{X_1, ..., X_m\}$, extracted features $\boldsymbol{\Psi}(\boldsymbol{X})$ and the target outputs of the i-th task $\boldsymbol{Y_i} = \{Y(X_1), ..., Y(X_m)\}$, the black-box classifier provides an internal learning function $f^i_{learn}(\boldsymbol{\Psi}(\boldsymbol{X}), \boldsymbol{Y_i}; \theta_i)$ that optimizes the classifier parameters θ_i to obtain the optimal mapping from the input features to the output labels for the training data. Once the parameters are learned, given a new sample X_{m+1} with features $\Psi(X_{m+1})$, the black-box classifier returns an output Y_{m+1} according to its internal inference function $f^i_{infer}(\Psi(X_{m+1}), Y_{i,m+1}; \hat{\theta}_i)$. For the i-th task, the optimal parameters (learning) and the optimal output (inference) are obtained by solving Equations 3.1 and 3.2 respectively.

$$Learning: \qquad \hat{\theta}_i = \underset{\theta_i}{\text{optimize}} \quad f^i_{learn}(\boldsymbol{\Psi}(\boldsymbol{X}), \boldsymbol{Y_i}; \theta_i) \tag{3.1}$$

$$Inference: \qquad \hat{Y}_i = \underset{Y_i}{\text{optimize}} \quad f^i_{infer}(\Psi(X), Y_i; \hat{\theta}_i) \tag{3.2}$$

The usage of black-box classifiers allows us to use and combine a great variety of classifiers which are known to perform well at their specific subtask without changing their inner structure. This allows us to merge them into a single unified model, which is able to exploit the information generated by each subtask to reinforce each other's beliefs and aid holistic scene understanding.

Notation: In order to formally and coherently describe the multiple instantiations of CCMs, we firstly introduce the notation used in this work. Consider n constituent tasks, with corresponding classifiers C_i $\forall i \in \{1, ..., n\}$, then:

$\Psi_i(X), \mathbf{\Psi}(X)$	$\Psi_i(X)$ indicates the features corresponding to the i-th task extracted from image X. $\mathbf{\Psi}(X)$ corresponds to the extracted features from X for all the tasks.
$\Gamma_i, \mathbf{\Gamma}$	Γ_i indicates the dataset for the i-th task consisting of m-length set of pairs $\{X_j, Y_j\}$. $\mathbf{\Gamma}$ represent the entire set of labeled data in the CCM.
$\Gamma_{X,i},\ \Gamma_{Y,i}$	The expressions $\Gamma_{X,i}$ and $\Gamma_{Y,i}$ are utilized to refer to the set of images and ground-truth labels of the i-th task respectively. Likewise $\mathbf{\Gamma}_X$ and $\mathbf{\Gamma}_Y$ refer to the entire set of labeled data.
$Y_i^{gt,l},\ \boldsymbol{Y}^{gt,l}$	$Y_i^{gt,l}$ corresponds to the ground-truth labels for the classifier C_i in the l-th layer. $\boldsymbol{Y}^{gt,l}$ indicates the set of ground-truth labels of all classifiers in the l-th layer $\{Y_1^{gt,l}, ..., Y_n^{gt,l}\}$.
$Y_i, \boldsymbol{Y}$	Y_i corresponds to the set of possible outputs of the classifier C_i. Note that Y_i is not instantiated for each layer, since the possible answer set is kept equal for each layer. In the case of Scene Classification for K classes, Y_{SC} corresponds to the label set $\{1, ..., K\}$. $\boldsymbol{Y}$ indicates the set of possible outputs for all the classifiers $\{Y_1, ..., Y_n\}$.
θ_i^l, θ^l	θ_i^l indicates the set of possible parameters corresponding to the classifier C_i^l. θ^l indicates the set of possible parameters for all the classifiers in the layer l, $\{\theta_1^l, ..., \theta_n^l\}$.
$\hat{Y}_i^l, \hat{\boldsymbol{Y}}^l$	$\hat{Y}_i^l$ corresponds to the output of the classifier C_i in the l-th layer. $\hat{\boldsymbol{Y}}^l$ indicates the set of outputs of all the classifiers of the l-th layer $\{\hat{Y}_1^l, ..., \hat{Y}_n^l\}$.
$\hat{\theta}_i^l, \hat{\theta}^l, \Theta$	θ_i^l indicates the learned parameters corresponding of the classifier C_i^l. $\hat{\theta}^l$ indicates the learned parameters of all the classifiers in the l-th layer $\{\hat{\theta}_1^l, ..., \hat{\theta}_n^l\}$. Θ compacts the learned parameters of all the classifiers in all layers into a single term.
$f_{learn}^{l,i}, f_{infer}^{l,i}$	$f_{learn}^{l,i}$ and $f_{infer}^{l,i}$ represent the learning and infer functions for the i-th task in the l-th layer.

With the necessary notation and definitions in place, we now describe the multiple CCM structures proposed by Heinz et. al. [Heitz 2009], Yi et. al. [Li 2012] and by us.

3.2 Cascaded Classification Models

Based on the general definition of the Cascaded Classification Models provided in the Chapter 3.1, we now formally define a L-tier CCM [Heitz 2009].

***L*-CCM:** An L-tier Cascaded Classification Model (L-CCM) is a cascade of classifiers of the target labels $\boldsymbol{Y} = \{Y_1, ..., Y_n\}^L$($L$ "copies" of each label) consisting of **independent** classifiers C_i^1 such that $f_{infer}^{1,i}(\Psi_i(X), Y_i^{gt,1}; \hat{\theta}_i^1)$, i.e. classifiers that do not include responses of other layers into their inference function, which are located in the first layer, and series of **conditional** classifiers $C_i^l \quad \forall l \in \{2, ..., L\}$ such that $f_{infer}^{l,i}([\Psi(X), \hat{\boldsymbol{Y}}_{-i}^{l-1}], Y_i^l; \hat{\theta}_i^l)$, i.e. classifiers that include the responses of earlier layers into their inference function. $\boldsymbol{Y}_{-i}^l$ indicates the assignment to all labels in the layer l *other than* Y_i^l. The labels $\boldsymbol{Y}^L$ at the final tier represent the final classification outputs. The structure of a L-CCM is depicted in the Figure 3.1. Note that Fig. 3.1 defines the input of the l-th layer as $\Upsilon_i(\Psi_i(X), \hat{\boldsymbol{Y}}_{-i}^{l-1})$. The union operator Υ_i defines a generalization over the concatenation operator [...] utilized in [Heitz 2009], which comes in handy in posterior instantiations of the model (See Chapter 3.4).

3.2.1 Inference

A CCM uses L classifier instantiations for each task, stacked into tiers as depicted in Figure 3.1. The i-th task instantiation of the first layer, referred to as "independent instantiation", learns exclusively utilizing the image features $\Psi_i(X)$ as input. Subsequent tiers of classifiers augment the image features with the outputs of the models in the preceding tier $\boldsymbol{Y}^{l-1}$. For a novel test instance, classification is performed by predicting the most likely MAP assignment to each of the variables in the final tier. The inference algorithm is given in Algorithm 1.

3.2.2 Learning

The CCM of Heitz et. al. learns in a feed-forward manner. The Learning Algorithm starts from the top level, training the independent classifiers first by maximizing classification performance on the training data set alone $\{\Psi_i(\Gamma_{X,i}), \Gamma_{Y,i}\}$. Because each classifier possesses an incorporated learning function $f_{learn}^{l,i}$, only the training dataset Γ_i that possesses ground-truth labels for that task is supplied into the learning function. In order to train each classifier C_i in the subsequent layer l of the CCM, we perform inference on the already trained classifiers in the $(l-1)$-tier of the CCM. From these output assignments $\hat{\boldsymbol{Y}}^{l-1}$, each classifier computes a new

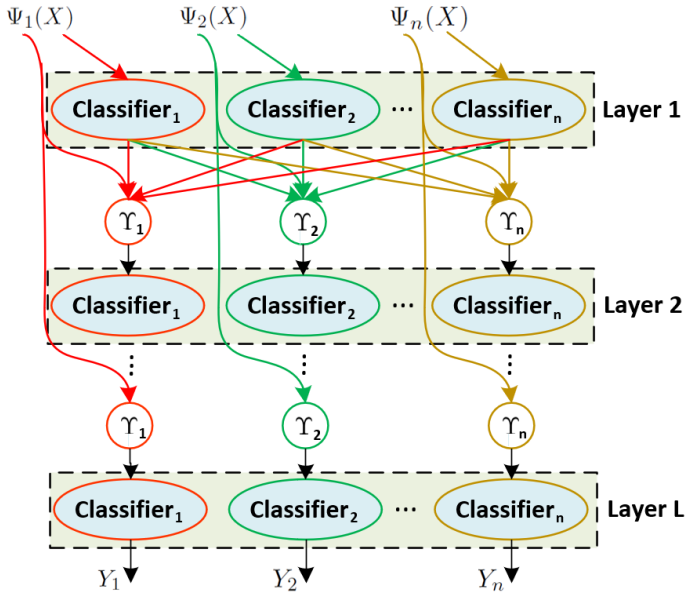


Figure 3.1: Cascaded Classification Model (CCM) Structure for combining n related classifiers ordered in L layers. Illustration based on [Heitz 2009]. $\Psi_i(X)$ $\forall i \in \{1, 2, ..., n\}$ corresponds to the first layer feature vector of the classifier C_i related to the i-th task. The outputs of the classifiers of the k-th layer are concatenated to the features of the classifiers of the $k+1$-th layer. The output of the CCM corresponds to the labels generated in the L-th layer for all the constituent tasks. Note that the classifiers C_i related to the i-th task might differ across layers.

set of features $[\Psi_i(X), \boldsymbol{Y}_{-i}^{\hat{l}-1}]$ and executes its corresponding learning algorithm $f_{learn}^{l,i}$. Note that the training regime involves using ground-truth information at every stage. This training schema is equivalent to train every layer independently, aiming to obtain the best possible classification performance on the ground truth labels. The learning algorithm is given in Algorithm 2.

Heitz et. al. provided two instantiations of the CCM framework, namely 2-CCM and 5-CCM, where X-CCM corresponds to a CCM instantiation with X layers. During their experiments, Heitz et. al. concluded that just by incorporating communication between the tasks (2-CCM) they achieved substantial improvement in all the involved tasks in comparison with their isolated solution. Moreover, they analyzed the influence of the depth in the CCM. In their experiments, they concluded that the depth of the CCM is directly related with an increasement of the model complexity and with a higher tendency to overfitting. Furthermore, due to the limits in the context signal, it cannot be expected to get unlimited improvements. Heitz et al. reported a general performance improvement of 7% with the 2-CCM instantiation without performance degeneration in any of the constituent tasks. By aggregating 3 additional layers (5-CCM), an additional 1% performance increase was obtained. Unfortunately, the performance improvement of 5-CCM over the 2-

Algorithm 1: CCM Inference Algorithm

Data: Input image X
Result: Labels $\boldsymbol{Y}$ of X for all the tasks

begin
  for $i \in \{1, ..., n\}$ **do**
    Calculate $\Psi_i(X)$ for the i-th task
    $\hat{Y}_i^0 \longleftarrow \emptyset$
// For all layers
for $l \in \{1, ..., L\}$ **do**
  // For all classifiers
  for $i \in \{1, ..., n\}$ **do**
    // Calculate the labels of the classifiers
    $\hat{Y}_i^l \longleftarrow$ optimize $_y$ $f_{infer}^{l,i}([\Psi_i(X)\ \hat{\boldsymbol{Y}}_{-i}^{l-1}], Y_i; \hat{\theta}_i^l)$
    $\hat{\boldsymbol{Y}}^l \cup \hat{Y}_i^l$
// Return the inferred labels at the last layer
return $\hat{\boldsymbol{Y}}^L$

Algorithm 2: CCM Learning Algorithm

Data: Input Dataset Γ
Result: Optimized parameters Θ for all classifiers in the model

begin
  for $i \in \{1, ..., n\}$ **do**
    Calculate $\Psi_i(X)$ for the i-th task
  Θ, $\boldsymbol{Y}^0$, $\hat{\theta}^0 \longleftarrow \emptyset$
// For all layers
for $l \in \{1, ..., L\}$ **do**
  // For all classifiers
  for $i \in \{1, ..., n\}$ **do**
    // Obtain the parameters $\hat{\theta}_i^l$ for the i-th task:
    $\hat{\theta}_i^l \longleftarrow$ optimize $_{\theta_i}$ $f_{learn}^{l,i}([\boldsymbol{\Psi}_i(\Gamma_{X,i})\ \hat{\boldsymbol{Y}}_{-i}^{l-1}], \Gamma_{Y,i}; \theta_i^l)$
    // Run inference over each classifier
    $\hat{Y}_i^l \longleftarrow$ optimize $_y$ $f_{infer}^{l,i}([\boldsymbol{\Psi}_i(\Gamma_{X,i})\ \hat{\boldsymbol{Y}}_{-i}^{l-1}], Y_i^l; \hat{\theta}_i^l)$
    $\hat{\boldsymbol{Y}}^l \cup \hat{Y}_i^l$
  $\Theta \cup \hat{\theta}^l$
// Return the list of parameters for all the model classifiers
return Θ

CCM was not consistent for all tasks. This behavior already demonstrates a grade of overfitting of the model on the available context signal. Additionally, 5-CCM has a substantial additional computational cost in comparison to 2-CCM. While for a 2-tier CCM structure training and inference are required over $2n$ and $n+1$ classifiers respectively, on a 5-tier CCM the requirements increase to $5n$ ($\approx +150\%$) and $4n+1$ ($\approx +300\%$) respectively. For additional information about the experiments and obtained results, we refer the reader to [Heitz 2009]. From their experiments, Heitz et. al. concluded that the 2-CCM is better suited for further research. This instantiation was posteriorly utilized by Li et. al. to construct their FE-CCM.

3.3 Feedback Enabled Cascaded Classification Models

Li et. al. [Li 2012] alleviated some of the most relevant problems of the CCM architecture. The inclusion of a feedback interface between layers strongly increased the learning capability of the algorithm, generating substantial performance improvements (Figure 3.2). The inference and learning algorithms of the FE-CCM are described hereafter. Note that in contrast with the CCM, the FE-CCM is strictly composed of 2 layers (2-CCM). In the following sections, the problems of inference and learning on a unique dataset containing labels for all the tasks is approached. Subsequently, we present the modifications made by Li et. al. to enable learning on disjoint data as well as some additional design choices defined by us, which are likewise utilized in the EFE-CCM.

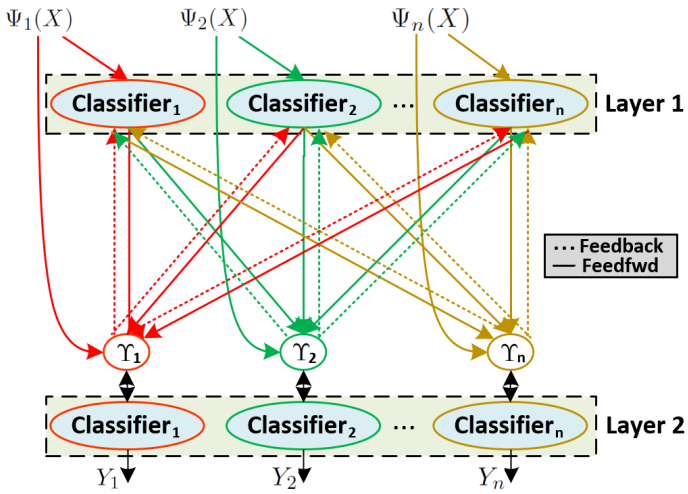


Figure 3.2: Structure of the Feedback Enabled Cascaded Classification Model (FE-CCM). Illustration based on [Li 2012]. In contrast to the structure of the CCM (Figure 3.1), FE-CCM provides a feedback interface from the later layer to achieve joint learning and optimization across layers. Note that the union operator Υ_i is equally defined as in CCM (See Chapter 3.2).

3.3.1 Inference

During inference, the final outputs $\hat{\boldsymbol{Y}}^2$ are deduced for a given input image X. As well as by the CCM, classification is performed by predicting the most likely MAP assignment to each of the variables in the final tier utilizing the learned parameters Θ. Formally, we perform:

$$\hat{Y}_i^1 = \underset{Y_i}{\text{optimize}} \quad f_{infer}^{1,i}(\Psi_i(X), Y_i^1; \hat{\theta}_i^1) \tag{3.3}$$

$$\hat{Y}_i^2 = \underset{Y_i}{\text{optimize}} \quad f_{infer}^{2,i}([\Psi_i(X) \quad \hat{\boldsymbol{Y}}^1], Y_i^2; \hat{\theta}_i^2) \tag{3.4}$$

Note that in contrast to the CMM of Heitz et. al. [Heitz 2009], Yi et. al. integrate the response of the classifier instantiation of the same task in the early layer in the layer-wise classification response ($\hat{\boldsymbol{Y}}^l$ in Equation 3.4 instead of $\hat{\boldsymbol{Y}}^l_{-i}$ in Algorithm 1, line 7). This is due to the fact that the ground-truth labels of the first layer $\boldsymbol{Y}^{gt,1}$ are able to change during training and the addition of the labels $Y_i^{gt,1}$ enables the algorithm to pass on information captured in early layers. For more information we refer the reader to the Chapters 3.3.2 and 3.3.3. Besides the aforementioned fact, the inference algorithm is identical (Algorithm 1 with $L = 2$).

3.3.2 Learning

In contrast to the learning approach of Heiz et. al., FE-CCM is not learn exclusively in a feed-forward manner. Here, the information is sent backwards to increase the learning capability and improve inference performance. During the training stage, the inputs $\Psi_i(X)$ as well as the target outputs, $Y_1^{gt,2}, Y_2^{gt,2}, ..., Y_n^{gt,2}$ of the second tier of classifiers are observed (since $Y_i^{gt,2} = \Gamma_{Y,i}$). In this learning framework, the variables $Y_1^{gt,1}, Y_2^{gt,1}, ..., Y_n^{gt,1}$ (outputs of the first layer into the second layer) are considered as **hidden variables**. Heitz et. al. [Heitz 2009] assumed independency between layers and thrived for each layer to produce the best output possible regarding the training set, without considering inter-layer effects. Under these assumptions, they utilize the ground-truth labels $\Gamma_{Y,i}$ for the training of the first layer as well. On the other hand, Li et. al. consider inter-layer dependency, shifting the objective from producing the best possible output in each layer to optimize the classification performance of the second layer as much as possible. Under this idea, the first-layer classifiers do not need to perform their best with regard to $\Gamma_{Y,i}$ but rather focus on error modes that would result in the second layer's output $\hat{\boldsymbol{Y}}^2$ being more accurate. In order to do so, the model is learned through an iterative Expectation-Maximization (EM) formulation considering the independence assumptions between classifiers represented in Fig. 3.2. In the feed-forward step, the variables $Y_i^{gt,1}$ are assumed to be known, with which we learn the classifier parameters Θ. During the feedback step the learned parameters Θ are utilized to find the best assignment to the first layer ground-truth labels $Y_i^{gt,1}$, such that the second layer is as accurate as possible. Since the variables $Y_i^{gt,1}$ are

no longer subjected to the ground-truth labels $\Gamma_{Y,i}$, as the iterations advance, the first layer classifiers start focusing on error modes that improve the performance in the second layer of classifiers.

Initialization: As depicted in the related work section (Chapter 2.2), maximum likelihood optimization algorithms usually build on top of hill-climbing-like optimization algorithms. Usually, these algorithms suffer of local optima and might find a suboptimal point in the probabilistic space. In order to alleviate this, Li et. al. provide a good initialization step to avoid optimization stagnation. The model is initialized by setting the latent variables $Y_i^{gt,1}$ to the ground-truth labels $\Gamma_{Y,i}$. This initial assignment assures that any variance in the hidden layer labels stays close to realistic assignments. Note that if we were to hold the assignment $Y_i^{gt,1} = \Gamma_{Y,i}$ during the training stage, the model would be equivalent to the 2-CCM in [Heitz 2009].

Feed-forward Step: In this step, the model parameters Θ are estimated. At this point, we assume that both $Y_i^{gt,1}$ and $Y_i^{gt,2}$ are known. This assumption holds, since $Y_i^{gt,1}$ is initialized to $\Gamma_{Y,i}$ and subsequently determined in the feedback step, and $Y_i^2 = \Gamma_{Y,i}$ during the entire learning process. The parameters θ_i of each classifier are learned independently utilizing the corresponding learning strategy of each "black-box" classifier as stated in Equation 3.1. Formally,

$$\hat{\theta}_i^1 = \underset{\theta_i^1}{\text{optimize}} \quad f_{learn}^{1,i}(\mathbf{\Psi}_i(\Gamma_{X,i}), Y_i^{gt,1}; \theta_i^1) \tag{3.5}$$

$$\hat{\theta}_i^2 = \underset{\theta_i^2}{\text{optimize}} \quad f_{learn}^{2,i}([\mathbf{\Psi}_i(\Gamma_{X,i})\ \hat{\boldsymbol{Y}}^1], Y_i^{gt,2}; \theta_i^2) \tag{3.6}$$

Note that for a unique data set that contains labels for all the considered tasks, $\Gamma_{X,i} = \Gamma_X\ \forall i \in \{1, ..., n\}$.

Feedback Step: In the feedback step, we are interested in inferring the values of the variables Y_i^1 that better describe the parameters θ_i found in the last step. This feedback step is the crux that provides information back to the first-layer classifiers regarding the error modes that are affecting the classification performance downstream the most. Given fixed values of Θ, we want to obtain assignments to the variables Y_i^1 that are good predictions from the first layer classifiers and simultaneously help to diminish classification errors of the variables $\hat{Y}_i^2$ as much as possible. In order to achieve this, Eq. 3.7 is solved for each instance X in Γ_X. Here, J_i^l evaluates the quality of the assignments at each layer.

$$\underset{\boldsymbol{Y}^{gt,1}}{\text{optimize}} \quad \sum_{i=1}^{n} (J_i^1(\Psi_i(X), Y_i^1; \hat{\theta}_i^1) + J_i^2([\Psi_i(X)\ \boldsymbol{Y}^1], Y_i^2; \hat{\theta}_i^2)) \tag{3.7}$$

One valid option is to define $J_i^1(\Psi_i(X), Y_i^1; \hat{\theta}_i^1)$ and $J_i^2([\Psi_i(X)\ \boldsymbol{Y}^1], Y_i^2; \hat{\theta}_i^1)$ as $f_{infer}^{1,i}$

and $f_{infer}^{2,i}$ respectively. The updated $\boldsymbol{Y}^{gt,1}$ are subsequently used to relearn the parameters Θ in the feed-forward step of the following iteration. Note that the updated $\boldsymbol{Y}^{gt,1}$ have continuous values. If the internal learning function of a classifier exclusively accepts labels, the values of $\boldsymbol{Y}^{gt,1}$ are thresholded to obtain labels. The learning algorithm of the FE-CCM is provided in Algorithm 3.

Algorithm 3: FE-CCM Learning Algorithm

Data: Input Dataset Γ, max Iterations I_{max}
Result: Optimized parameters Θ for all classifiers in the model

begin
  Initialize $\boldsymbol{Y}_0^{gt,1}$ with Γ_Y
  $it \longleftarrow 1$
  Convergence $\longleftarrow$ *false*
while $(it \leq I_{max}) \vee (Convergence = false)$ **do**
  `// Fix` $\boldsymbol{Y}_{it-1}^{gt,1}$ `and estimate` Θ `(Equations 3.5 and 3.6)`
  $\Theta \longleftarrow$ `FeedforwardStep(`$\boldsymbol{Y}_{it-1}^{gt,1}$`)`
  `// Fix the parameters` Θ `and estimate` $\boldsymbol{Y}_{it}^{gt,1}$ `(Equation 3.7)`
  $\boldsymbol{Y}_{it}^{gt,1} \longleftarrow$ `FeedbackStep(`Θ`)`
  if $(\boldsymbol{Y}_{it}^{gt,1} = \boldsymbol{Y}_{it-1}^{gt,1})$ **then** Convergence $\longleftarrow$ *true*
`// Return the list of parameters for all the model classifiers`
return Θ

3.3.3 Training with Heterogeneous Data Sets

Often available data sets are disjoint for multiple tasks, i.e. data samples often do not possess labels for multiple tasks. Li et. al. incorporated the ability to learn on heterogeneous for the general case, where Γ_i exclusively holds labels for the i-th task. In this section, the required modifications in the feed-forward and the feedback steps while learning on disjoint datasets are provided.

Feed-forward Step: During the feedback step, labels $\boldsymbol{Y}^{gt,1}$ are generated for the entire dataset Γ. In other words, we generate labels for all the data samples in each of the disjoint data sets for all the considered tasks. Although the generated labels are not perfect, they are useful to provide supplementary inter-task knowledge into the first layer. Following this line of thinking, Li et. al. utilize all the data sets in order to relearn each of the first-layer classifiers. If the internal learning function of the black-box classifier is additive over the data points, the training procedure corresponds to:

$$\hat{\theta}_i^1 = \underset{\theta_i^1}{\text{optimize}} \quad \sum_j \sum_{X \in \Gamma_j} \pi_j \; f_{learn}^{1,i}(\Psi_i(X), Y_i^{gt,1}; \theta_i^1) \tag{3.8}$$

where π_j correspond to importance factors assigned to different data sets and satisfy $\sum_j \pi_j = 1$. Insight on appropriate selection of the importance factors π_j are provided in the Chapter 3.3.5. For cases where the internal learning function is not additive over the data points, one can randomly sample a subset of data $\boldsymbol{X}^j$ from each data set Γ_j, i.e. $\boldsymbol{X}^j \subseteq \Gamma_j$ to form a new data set $\boldsymbol{X}^* = [\boldsymbol{X}^1, ..., \boldsymbol{X}^n]$. Under this scheme, the importance factor π_j for each subset $\boldsymbol{X}^j$ is implicitly equivalent to the proportion of data samples extracted from the original dataset X^j in the new created data set $\boldsymbol{X}$, i.e. $\pi_j = \frac{|\boldsymbol{X}^j|}{|\boldsymbol{X}^*|}$, where $|\cdot|$ indicates the cardinality of the data set. In this fashion, the parameters of the first layer are learn solving:

$$\hat{\theta}_i^1 = \underset{\theta_i^1}{\text{optimize}} \quad f_{learn}^{1,i}(\Psi_i(\boldsymbol{X}^*), Y_i^{gt,1}; \theta_i^1) \tag{3.9}$$

In order to relearn the parameters of the second-layer classifiers, we exclusively utilize the data points that originally have ground-truth labels for the corresponding task, i.e. by solving Equation 3.6. The only difference with the feed-forward step in the second layer for a unique data set containing labels for all the tasks, is that here the dataset $\Gamma_{X,i}$ is no longer equivalent to Γ_X.

Feedback Step: In this step, we adjust the Equation 3.7 to exclusively consider the data points of the data set that contains ground-truth labels for the corresponding task. Since just the data set Γ_j contains labels for the j-th task, we only consider J_j^2 in the second term. However, since we generate labels to all the tasks for the samples in Γ_j during the feed-forward step, we consider all the $J_i^1 \quad \forall i \in \{1, ..., n\}$ in the first term. Formally, in order to obtain assignments for the values $Y_i^{gt,1}$ for each of the data samples X in Γ_j, we solve:

$$\underset{\boldsymbol{Y}^{gt,1}}{\text{optimize}} \quad \sum_{i=1}^{n} (J_i^1(\Psi_j(X), Y_i^1; \hat{\theta}_i^1) + J_j^2([\Psi_j(X)\ \boldsymbol{Y}^1], Y_j^2; \hat{\theta}_j^2)) \tag{3.10}$$

The feedback step in the FE-CCM is the most representative extension of the framework proposed by Li et. al. over the predecessor work of Heitz et. al. One important inconvenience that arises when interpreting and re-implementing the FE-CCM from [Li 2012] is caused due to the insufficient information provided by the authors regarding the specific implementation of the optimization problem solved during this step. This problem leads to an ambiguous interpretation of the feedback formulation utilized in their experiments, which simultaneously hampers any efforts to correctly compare their implementation with novel framework variations. In order to cope with this problem and to facilitate an easier comparison interface with future works, we provide concrete formulations of the optimization problems utilized in this work. Simultaneously, we aim to provide hints and suggestions for the incorporation of additional tasks in the framework, which we consider of high relevance, since the formulation of the optimization problem varies depending on the nature of the considered task.

3.3.4 Probabilistic Interpretation of the Feedback Step

The formulation of the optimization problem for the feedback step introduced in the previous section (Eq. 3.10) aims to find the assignments of all the ground-truth labels of the first layer for all the tasks $\boldsymbol{Y}^{gt,1} = \{Y_1^{gt,1}, Y_2^{gt,1}, ..., Y_n^{gt,1}\}$ that help the most to correct the predictions $\hat{Y}_j^2$ of the j-th task classifier in second layer, while being good predictions of the respective classifiers in the first layer. For the sake of clearness, we consider the heterogeneous training case during the derivation of the optimization problem formulations. In the homogeneous case, i.e. an unique dataset containing annotations for all the tasks, one simply needs to merge the parts related to the second layer classification from the heterogeneous case for all the considered tasks into a single optimization problem (Eq. 3.14).

Consider all the constituent classifiers in the model to produce a probabilistic output of the form $P(Y = y|X = x)$. In this sense, the Eq. 3.10 can be reformulated as

$$\max_{\boldsymbol{Y}^{gt,1}} \quad \log P(\hat{Y}_j^2 = Y_j^{gt,2}, \boldsymbol{Y}^{gt,1} | \Psi_j(X)) \tag{3.11}$$

i.e. as the joint appearing log-probability of all the labels $Y_i^{gt,1}$ from the first layer and the inference of the second layer classifier being correct $\hat{Y}_j^2 = Y_j^{gt,2}$. This expression can be further factorized utilizing the independencies represented in the directed model of Fig. 3.2 to obtain a formulation similar to Eq. 3.10, which can be solved performing MAP inference on the latent variables $\boldsymbol{Y}^{gt,1}$:

$$\max_{\boldsymbol{Y}^{gt,1}} \quad \log P(\hat{Y}_j^2 = Y_j^{gt,2} | \Psi_j(X), \boldsymbol{Y}^{gt,1}) + \sum_{i=1}^{n} \log P(\hat{Y}_i^1 = Y_i^{gt,1} | \Psi_i(X)) \tag{3.12}$$

An important additional situation appears when utilizing non-probabilistic classification frameworks such as Support Vector Machines (SVMs). In such cases, we recommend utilizing a (semi-)probabilistic transformation over the classifier's output, such as the subsequent fitting of a sigmoid function over the SVM response proposed by John Patt in [Platt 1999], and the normalized voting strategy, proposed by Jeróme Friedman in [Friedman 1996], for binary and multi-class classification, respectively.

As stated before, the formulation of the optimization problem in the feedback step requires a different formulation regarding the nature of the task. We identify two types of problems (Fig. 3.3): **global-based inference tasks**, were one singular inference process produces an unique label for the entire image (e.g. scene classification) and **local-based inference tasks**, were several "independent" labels are generated by multiple inference processes over several (independent) regions in the image, e.g. object recognition, saliency detection, depth estimation, ... Note that in tasks such as depth estimation, regions can be defined as singular pixels. For **global-based inference tasks**, the first term of Eq. 3.12 corresponds to a single

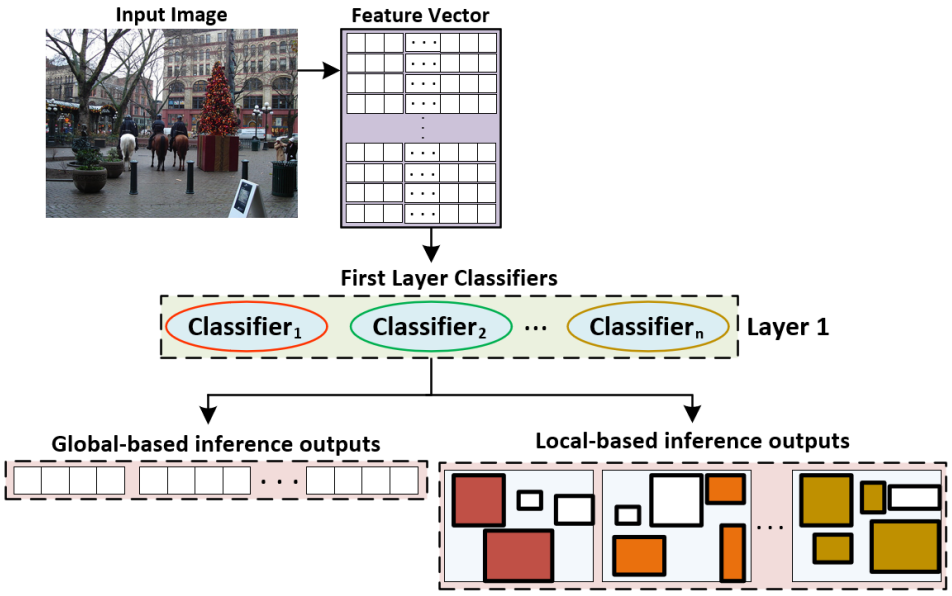


Figure 3.3: First layer output comparison of global-based and local-based inference tasks. Consider the set of input feature vectors for the first layer $\mathbf{\Psi}(X) = [\Psi_i(X)\ \forall i \in \{1, 2, ..., N\}]$ extracted from an input image X. The first layer classifiers generate a set of outputs $\hat{Y}^1$, which can be divided in two disjoint sets G, L composed of global-based and local-based inference tasks respectively. Note that while the set G is composed of several 1D-vectors, usually indicating the probability of the image X belonging to a global class, the set L is composed of multiple 2-dimensional maps, which usually describe the likelihood of a region in the image X to belong to a local class, such as an object or a segment. Filled and unfilled regions indicate regions classified as positive and negative respectively.

probability, e.g. the probability of the image belonging to a certain class. In this scenario, the formulation provided in Eq. 3.12 is complete. In other words, we seek the best combination of labels for all tasks in the first layer that maximizes the probability of the image belonging to the correct label in the second layer. Conversely, for **local-based inference tasks**, the first term in Eq. 3.12 does not longer correspond to a single probability but rather to the agglomeration of the probability of all the regions in an image taking a certain label at once. Formally, the optimization problem is reformulated as:

$$\max_{\boldsymbol{Y}^{gt,1}} \quad \sum_{\lambda=1}^{\Lambda} \log P(\hat{Y}^2_{j,\lambda} = Y^{gt,2}_{j,\lambda} | \Psi_j(X), \boldsymbol{Y}^{gt,1}) + \sum_{i=1}^{n} \log P(\hat{Y}^1_i = Y^{gt,1}_i | \Psi_i(X)) \quad (3.13)$$

where Λ describes the number of regions in the image. In other words, we seek the best combination of labels for all tasks in the first layer that maximizes the joint probability of all the regions in the image being correctly classified in the second layer. For local-based inference tasks in the first layer, the interpretation is coequal.

For the sake of completeness, we provide an exemplary formulation for learning on homogeneous datasets containing both global- and local-based inference tasks. Consider G and L, with $\dim G + \dim L = n$, the set of global- and local-based inference tasks respectively. The obtained optimization problem formulation is shown in Eq. 3.14. Note that in this case, the feature vector $\Psi_{l,\lambda_l}(X)$ becomes different for each region in the image.

$$\max_{\boldsymbol{Y}^{gt,1}} \quad \sum_{g=1}^{G} \left[\log P(Y_g^{gt,2} | \Psi_g(X), \boldsymbol{Y}^{gt,1}) + \log P(Y_g^{gt,1} | \Psi_g(X)) \right] + \sum_{l=1}^{L} \sum_{\lambda_l=1}^{\Lambda_l} \left[\log P(Y_{l,\lambda_l}^{gt,2} | \Psi_{l,\lambda_l}(X), \boldsymbol{Y}^{gt,1}) + \log P(Y_{l,\lambda_l}^{gt,1} | \Psi_{l,\lambda_l}(X)) \right] \quad (3.14)$$

3.3.5 Selecting importance factors: FE-CCM Instantiations

The selection of the importance factors π_j introduces an additional parameter into the model. By varying the selected importance factors, multiple instantiations of the FE-CCM are generated, each of which holds different properties for a particular task. In the following paragraphs, we introduce the multiple instantiations proposed by Li et. al. in [Li 2012].

- **Unified FE-CCM:** In this instantiation, the general goal of the model is to achieve the maximal possible improvements in all tasks simultaneously with a single set of parameters Θ. In order to do so, it is desired to balance the data from different data sets Γ_j. Towards this goal, π_j is set to be inversely proportional to the cardinality of the data set Γ_j for the j-th task. The unified FE-CCM balances the amount of data in different data sets, based on the Equations 3.8 or 3.9 depending on the nature of the classifiers.

- **One-goal FE-CCM:** In this instantiation, the importance factor π_j is set to one for the corresponding task and to null otherwise, i.e. $\pi_j = 1$ if $j = k$ and 0 otherwise. This is an extreme configuration to favor the k-th task. In this case, the training of the first-layer classifiers retain feedback information exclusively from the classifier C_k^2. In other words, the model only utilizes the data set that contains labels for the k-th task Γ_k. Although the goal of this setting is to completely benefit the k-th task, Li et. al. demonstrated that this configuration often results in overfitting and does not even achieve the best results in comparison with other instantiations. This is due to a hard degeneration of the model occasioned by: (1) a relative small data set in comparison with other instantiations (exclusive consideration of the data set Γ_j instead of Γ during training) and (2) the lack of feedback from the remaining tasks in the second layer into the first one (the model is only able to gain information back from the second layer classifier of the goal task). In

this case, for a FE-CCM involving n tasks, n different models are trained to obtain n independent Θ configurations, one for each particular task.

- **Target-specific FE-CCM:** This instantiation aims to optimize the performance of a particular task. However, in contrast to the one-goal FE-CCM instantiation, where the classifiers of the remaining tasks in the second layer are removed, in this instantiation all the tasks in the second layer are conserved. Here, the parameters π_j are determined by data-driven selection. Specifically, π_j are selected trough cross-validation on a hold-out validation set in the learning process in order to optimize the second-layer output of a specific task. Since Target-specific FE-CCM retains all the classifiers in the second layer, the first layer is able to obtain feedback information from all these classifiers, resulting in a more robust and accurate instantiation than One-goal FE-CCM. As well as by One-goal FE-CCM, we require n different models to obtain n independent Θ configurations, one for each target task.

In the experiments carried out by Li et. al. in [Li 2012], they perform comparisons between the task baselines, an all-features-direct classifier and the multiple FE-CCM instantiations presented before. Here, the all-features-direct classifier corresponds to a classifier that takes all the features of all subtasks, appends them together and builds a separate classifier for each task. During their experiments they concluded that all the FE-CCM instantiations had substantial performance improvements in relation to the baselines and the all-features-direct classifiers. From these results, one can easily conclude that the performance improvement of the FE-CCM is not solely related to an increasement in the feature vector of the corresponding tasks but rather due to the inclusion of more contextual and interrelational information between tasks, which helps for further improvement of the classification performance. During the comparison of the three FE-CCM instantiations, the target-specific FE-CCM performed the best. This was expected as the importance factors are adequated to the specific target task while conserving a broad feedback spectrum from the second layer classifiers. Although the target-specific FE-CCM performs slightly better that the unified FE-CCM ($\approx +0.3\%$ over all considered tasks), it is worth highlighting that the unified FE-CCM achieves this with a single set of parameters Θ optimized for all the tasks simultaneously. From the task complexity point of view of the multiple instantiations, the training scheme of the target-specific instantiation requires to train n independent FE-CCMs, one for each classification task, while the unified FE-CCM instantiation requires training of a single model. During an inference process, to generate outputs for all the considered tasks, the target-specific FE-CCM requires inference over all the first-layer classifiers and the task-specific classifier in the second layer for each adjusted target-specific FE-CCM, specifically, it requires inference over $n(n+1)$ classifiers, while in the unified FE-CCM this is achieved by conducting inference over $2n$ classifiers.

Li et. al. also compared the FE-CCM with the CCM of Heinz et. al. In order to do so, they trained both the FE-CCM and the CCM model on the DS1 data set of [Heitz 2009], which contains ground-truth labels for scene recognition and object detection. They performed experiments over two settings in this dataset: (1) training over the fully labeled data and (2) training only with scene labels for one half of the training data and with only the object labels for the second half. FE-CCM outperformed the CCM for both experimental settings. Surprisingly, the FE-CCM trained on partially labeled data was already able to outperform the CCM trained on fully labeled data. This indicates that the improvement achieved by the FE-CCM is not solely due to the generation of more labels for training the first-layer classifiers, but also due to finding useful error modes in the first-layer classifiers. For further information about the experiments and obtained results, we refer the reader to [Li 2012].

Based on the results obtained by Li et. al., we conclude that the improvement obtained by the target-oriented FE-CCM does not justify the substantial time complexity increasement required both for training and inference. Therefore, we select the unified FE-CCM as baseline for the EFE-CCM.

3.4 Enhanced Feedback Enabled Cascaded Classification Models

The most representative extension of the EFE-CCM over the FE-CCM [Li 2012] is the incorporation of an inter-layered feature reduction approach. This feature reduction module aims to alleviate some of the most relevant problems of the FE-CCM architecture, mainly related to performance and complexity overhead over the baselines. Consider the input feature vector $\Psi_i(X)$ and $\Phi_i(X)$ of the i-th task for the first and the second layer respectively. Furthermore, consider a union operator Υ_i, which maps the input feature vector of the first layer $\Psi_i(X)$ and the output of the first layer classifiers $\hat{\boldsymbol{Y}}^1$ into a single feature vector $\Phi_i(X) = \Upsilon_i(\Psi_i(X), \hat{\boldsymbol{Y}}^1)$, which is subsequently fed into the i-th task classifier in the second layer. So far, the union operator Υ_i has been exclusively used to concatenate the first layer outputs $\hat{\boldsymbol{Y}}^1$ to the original feature vector $\Psi_i(X)$ of the i-th class (See Chapters 3.2, 3.3). Li et. al. [Li 2012] utilized multiple variations in the concatenation strategy as a function of the analyzed task. For example, while the entire detection maps obtained from the object recognition task are concatenated to the feature vector of some tasks (e.g. scene categorization), for others (e.g. saliency detection) a mere three-dimensional vector is added, where each element represents the average score for the corresponding pixel/patch/bounding-box [Li 2012]. Although the authors do not provide any explicit reasoning as of why particular variations are selected for different tasks, we argue that a very feasible reasoning could rely on a trade-off between the task complexity and the expected contribution of the first layer output in the discriminant power of the second-tier

classifier. Consider the saliency detection learning scheme of [Li 2012] and a feature detection map size of 16×16. Since the feature input of the first layer saliency detection task is composed of four elements, i.e. $\Psi_{SD}(X) \in \Re^4$, the included overhead of concatenating $4 * 16^2 = 1024$ additional features for the classifier in the second layer seems extremely disproportionate. Therefore, the concatenation of a much more compact version of the first layer output vector appears more reasonable.

We argue, however, that the definition of the union operator Υ_i utilized by Heitz et. al. [Heitz 2009] and subsequently by Li et. al. [Li 2012] is not optimal due to the following factors:

1. Since the classifier responses of the constituent tasks $\hat{\boldsymbol{Y}}$ in Holistic Scene Understanding are known to be (strongly) correlated (see Chapters 1, 3.1), it is intuitive to think that much of the provided information is superfluous, i.e that, by means of (non)linear dependency or statistical equivalence, equal pieces of information are repetitively included in $\Phi_i(X)$.

2. As the number of tasks in the first layer grows, so do the number of features that must be included in the second layer too. Consider two FE-CCMs including the Object Recognition task for n and N classes respectively. For $n < N$ and a standard prediction map size for all the classes of $W \times H$-pixels, the output of the object detection task in each FE-CCM amounts to $n(W*H)$ and $N(W*H)$ additional features respectively. This corresponds to a difference of $(N-n)(W*H)$ features between both FE-CCM instantiations. For large N, the FE-CCM classification performance inevitably degrades, due to the unbalanced growing between the number of training samples and the number of utilized features for learning (see Chapter 2.2.6).

3. CCM and FE-CCM requires the selection of multiple indispensable design parameters, whose evaluation and, furthermore, optimal selection is extremely difficult. As a matter of fact, some of them were neither defined nor specified by Li et. al. [Li 2012] or by Heitz et. al. [Heitz 2009] in their corresponding works. One very representative example is the definition of the standard object detection map size in the Object Recognition Task, which is indispensable to obtain a constant feature vector size for subsequent classification tasks. We consider the leak of systematic means for the appropriate selection of these design parameters as a disadvantage of the CCM and the FE-CCM.

A more appropriate selection of the union operator Υ_i should utilize a sparse feature space transformation, such that the input feature vector of the second layer $[\Psi_i(X), \hat{\boldsymbol{Y}}^1]$ is mapped into a (relatively) low-dimensional feature vector $\Phi_i(X)$, such that superfluous information is automatically detected and removed from the feature vector. Such a mapping can be achieved by implementing a feature reduction algorithm over the first layer feature vector $\Psi_i(X)$ and the output of the first layer classifiers $\hat{\boldsymbol{Y}}^1$. Note that a functional f_{red} could be defined to evaluate

the suitability of the feature reduction transformation by, for example, measuring the explained variance of the resulting space with a penalization on the model complexity, allowing for an understandable and practical selection strategy of the optimal design parameters. In the scope of this work, we introduce two feature extraction strategies into the model:

(PCA)FE-CCM: Here, we initially concatenate the first-layer output $\hat{\boldsymbol{Y}}^1$ to the original feature vectors $\Psi_i(X)$ in the same manner as CCM and FE-CCM. Subsequently, we perform Principal Component Analysis (PCA) and select the n first Principal Components (PC), such that the explained variance of the transformation surpasses an specified threshold t_Σ. Formally, we solve the optimization problem stated in Eq. 3.15, where λ_{PCA} represents the vector containing the eigenvalues pursuant to the eigenvectors utilized for the PCA transformation and λ_Σ symbolize the total variance of the training data set in the original feature space.

$$\begin{aligned} \min_n \quad & f_\Sigma(n) = \frac{1}{\lambda_\Sigma}\big[\sum_{i=1}^{n} \lambda_{\mathrm{PCA}}(i)\big] \\ \text{s.t.} \quad & f_\Sigma(n) \geq t_\Sigma \end{aligned} \tag{3.15}$$

The resulting n Principal Components are utilized for the classification task in the second layer. Formally, the input feature vector of the second layer $\Phi_i(X)$ is defined as:

$$\Phi_i(X) = \mathrm{PCA}\left([\Psi_i(X), \hat{\boldsymbol{Y}}^1], n\right) \tag{3.16}$$

(LDA)FE-CCM: For the second one, we initially divide the first-layer classifier responses regarding the nature of the task into global-based inference tasks and local-based inference tasks as introduced in Chapter 3.3.4. Consider G and L, the set of global- and local-based inference tasks respectively, such that $G \cup L = \hat{\boldsymbol{Y}}^1$ (Fig. 3.3). We split the feature reduction problem into an unsupervised feature reduction problem over the concatenated original feature vectors $\Psi_i(X)$ and the global-based inference task set G, $[\Psi_i(X), G]$ and a supervised feature reduction problem over the set of local-based inference tasks L. The results obtained from the disjoint feature reduction problems are subsequently concatenated to form the input feature vector of the second layer Φ_i. The unsupervised feature reduction problem is once more formulated utilizing Principal Component Analysis (PCA) (Eq. 3.15, 3.16), this time just for the feature subset form by the input feature vector of the first layer and those classifier responses, whose ground-truth labels do not follow any spatial distribution but rather are equal for the entire image (Eq. 3.17).

$$\Phi_{i,G}(X) = \mathrm{PCA}\left([\Psi_i(X), G], n\right) \tag{3.17}$$

On the other hand, the supervised feature reduction problem is formulated utilizing Linear Discriminant Analysis (LDA) on the local-based inference task set L. Here, we interpret the detection maps generated by the local-based inference tasks of

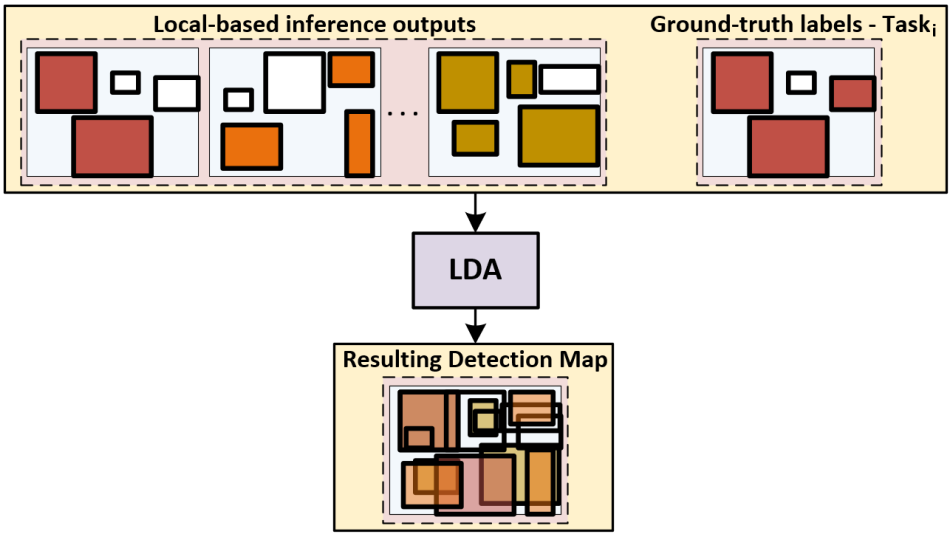


Figure 3.4: LDA procedure on the local-based inference outputs. Consider L the set of local-based inference outputs, composed of N $(W \times H)$-sized detection maps and $Y_i^{gt,1}$ the ground-truth labels of the i-th task. The LDA compresses the N detection maps into a unique detection map, which optimally describes the first label ground-truth labels of the i-th task $Y_i^{gt,1}$, subject to the assumptions of the LDA formulation (See Chapter 2.2.6). Note that the obtained detection map beholds spatial information after the LDA transformation. We explicitly utilize the spatial distribution of the transformed detection map in the training scheme of the following layer.

the first layer as multiple wavelength bands of the same image, related to a 2-Dimensional distribution of single pixel-wise defined labels (Fig. 3.4). Assume that the response of the local-based inference task set L is composed of N detection maps M_j, each with an standardized dimension of $W \times H$ pixels. Furthermore, assume we are interested in implementing Linear Discriminant Analysis for the i-th task, i.e. with consideration of the ground-truth labels $Y_i^{gt,1}$. Now, we generate $W * H$ {feature vector,label} pairs, where each feature vector is composed of N "wavelength values". The mathematical formulation of the supervised feature reduction problem is shown in Eq. 3.18, where (x, y) is intended to signalize the spatial distribution of the {feature vector,label} pairs. The input feature vector of the second layer $\Phi_i(X)$ obtained from the (LDA)FE-CCM is shown in Eq. 3.19.

$$\Phi_{i,L}(X) = \mathrm{LDA}\left([M_{j,(x,y)}\ \forall j \in \{1, ..., N\}], Y_{i,(x,y)}^{gt,1}\right) \tag{3.18}$$

$$\begin{aligned} \Phi_i(X) &= [\Phi_{i,G}(X), \Phi_{i,L}(X)] \\ &= \left[\mathrm{PCA}\big([\Psi_i(X), G], n\big), \mathrm{LDA}\big(L, Y_i^{gt,1}\big)\right] \end{aligned} \tag{3.19}$$

Note that with this formulation, the larger the number of available wavelength bands N, the better the LDA transformation will become. Hence, our formulation

do not suffer from a growing of the feature vector $\Psi_i(X)$ proportional to the number of constituent tasks in the EFE-CCM. In fact, we expect richer feature vectors $\Psi_i(X)$ with (roughly) the same dimension for larger N. Furthermore, the resulting input feature vector of the second layer $\Phi_{i,L}(X)$ beholds meaningful information about the spatial {feature vector,label} distribution over the image (Fig. 3.4). Based on several works from the field of hyperspectral imaging and hyperspectral feature reduction (See Chapter 2.2.6), we argue that the beheld spatial information is of great relevance for the classification problem. There is an important effect of utilizing a supervised approach over the L set. As the proposed formulation solely considers the labels of the i-th task, we need to construct N disjoint LDA transformations, one for each local-based inference task included in the second layer. This could be considered disadvantageous, due to the additional complexity added to the model, especially for large values of N. In order to evaluate this behavior, we perform multiple experiments to compare the execution times of the proposed EFE-CCM instantiations with those of the original FE-CCM.

With this approach we aim to unite the advantageous properties of both supervised and unsupervised feature reduction approaches into a single model. We argue that such a strategy could aid the feature reduction results, when the assumptions of one of the components is (slightly) violated. Furthermore, we argue that the consideration of the actual ground-truth labels in the feature reduction model helps towards especially well-designed feature transformations for the subsequent classification tasks. In order to evaluate this behavior, we perform experiments between the (PCA)- and the (LDA)FE-CCM instantiations and compare the added value of each strategy.

3.4.1 Inference

As well as by the inference process of the FE-CCM (Chapter 3.3.1), with the EFE-CCM we aim to infer the outputs of the second classification layer $\hat{\boldsymbol{Y}}^2$ for a given input image X, such that they correspond to the most likely MAP assignment of every task by means of the learned parameters Θ. The most relevant difference during the inference process with regard to that of FE-CCM is the generalization of the operator Υ_i in Eq. 3.4. Formally, we perform:

$$\hat{Y}_i^1 = \underset{Y_i}{\text{optimize}} \quad f_{infer}^{1,i}(\Psi_i(X), Y_i^1; \hat{\theta}_i^1) \tag{3.20}$$

$$\hat{Y}_i^2 = \underset{Y_i}{\text{optimize}} \quad f_{infer}^{2,i}(\Upsilon_i(\Psi_i(X), \hat{\boldsymbol{Y}}^1), Y_i^2; \hat{\theta}_i^2) \tag{3.21}$$

where $\Upsilon_i(X)$ corresponds to Eq. 3.16 and Eq. 3.19 for the (PCA)- and (LDA)FE-CCM instantiations respectively. Besides the aforementioned fact, the inference algorithm remains identical to that of the CCM (Algorithm 1 with $L = 2$).

3.4.2 Learning

The learning approach of the EFE-CCM is very similar to that of FE-CCM (Chapters 3.3.2, 3.3.3). The EFE-CCM is learned through an iterative Expectation-Maximization formulation as well. The differences between both approaches mainly appear in the **feed-forward step**, in which besides the parameters of the classifiers Θ, we learn the parameters of the introduced inter-layered feature reduction transformations $\hat{\Upsilon} = [\hat{\Upsilon}_i \ \forall i \in \{1, ..., n\}]$.

Feed-forward Step: In the feed-forward step, the variables $Y_i^{gt,1}$, $Y_i^{gt,2}$ are assumed to be known, with which we estimate the parameters of the classifiers Θ and the inter-layered feature transformations $\hat{\Upsilon}$. This assumption holds, since $Y_i^{gt,1}$, is either initialized to $\Gamma_{Y,i}$ or obtained via the feedback step, and $Y_i^{gt,2} = \Gamma_{Y,i}$ is kept constant during the entire learning process. The parameters θ_i are learned independently utilizing the corresponding learning strategy of each "black-box" classifier as stated in Chapter 3.1. On the other hand, the feature reduction transformations $\hat{\Upsilon}_i$ are obtained by optimizing a suitability function f_{red} over the transformation space for the input feature vector of the first layer $\Psi_i(X)$ and the response of the first layer classifiers $\hat{\boldsymbol{Y}}^1$. Formally, the feedforward step is formulated as follows:

$$\hat{\theta}_i^1 = \underset{\theta_i^1}{\text{optimize}} \quad f_{learn}^{1,i}(\boldsymbol{\Psi}_i(\Gamma_{X,i}), Y_i^{gt,1}; \theta_i^1) \tag{3.22}$$

$$\hat{\Upsilon}_i = \underset{\Upsilon_i}{\text{optimize}} \quad f_{red}(\boldsymbol{\Psi}_i(\Gamma_{X,i}), \hat{\boldsymbol{Y}}^1, Y_i^{gt,2}; \Upsilon_i) \tag{3.23}$$

$$\hat{\theta}_i^2 = \underset{\theta_i^2}{\text{optimize}} \quad f_{learn}^{2,i}(\hat{\Upsilon}_i(\boldsymbol{\Psi}_i(\Gamma_{X,i}), \hat{\boldsymbol{Y}}^1), Y_i^{gt,2}; \theta_i^2) \tag{3.24}$$

Note that we include the ground-truth labels $Y_i^{gt,2}$ of the second layer in the feature reduction formulation (Eq. 3.23). This signalizes that both supervised and unsupervised (with $Y_i^{gt,2} = \emptyset$) feature reduction methods can be adequated into the EFE-CCM formulation. In this work, we consider an unsupervised (PCA)- and an supervised (LDA)- instantiation, for which the suitability function is defined in Eq. 3.15 and Eq. 3.18 respectively. It is worth highlighting that both feature reduction strategies are dependent on the current iteration of the training algorithm, since the output of the first layer classifiers $\hat{Y}_i^1$ are iteration dependent as well. Consequently, we obtain a different solution for the Eq. 3.23 in every iteration.

To finalize, there is a last relevant fact that needs to be mentioned. Previously, we stated that the formulation of the EFE-CCM mainly has an impact on the feed-forward step when confronted with that of the FE-CCM. However, as the resulting second-layer features are not uniform across CCM instantiations, one needs to hold in mind that this new formulation indirectly affects the feedback step as well.

Chapter 4

Implementation Details

Contents

In this section, we describe the implementation details of the CCM, FE-CCM and EFE-CCM utilized for our experiments. Our implementation is performed in `C++` with help of the open-source OpenCV Library[1] and the Knitro Nonlinear Optimization Solver[2].[3]

In order to perform an accurate meaningful comparison, we utilize the same baselines for all CCM instantiations: CCM, FE-CCM, EFE-CCM. Our baseline selection follows that of Li et. al. in [Li 2012]. In order to provide a clear description of the implementation details, we utilize the notation introduced in Chapter 3.1. Let i be the index of the considered task. In this work we consider two tasks for our experiments on Holistic Scene Understanding: Scene Categorization ($i = 1$) and Object Recognition ($i = 2$). The inputs of the i-th task in the first layer are given by the extracted feature vector $\Psi_i(X)$, while those of the second layer are provided by the output of the union operator Υ_i over the first-layer input vector and the output of the first layer classifiers. The obtained feature vector is described as:

$$\Phi_i = \Upsilon_i(\Psi_i(X), \hat{Y}_1^1, \hat{Y}_2^1) \tag{4.1}$$

For the CCM and FE-CCM instantiations, the union operator Υ_i is defined as a concatenation operator. In this case the obtained feature vector is described as:

$$\Phi_i(X) = [\Psi_i(X)\ \hat{Y}_1^1\ \hat{Y}_2^1] \tag{4.2}$$

[1] `http://opencv.org/`
[2] `http://artelys.com/en/optimization-tools/knitro`
[3] Our implementation is available online at `http://github.com/RomeroGuDw/master-thesis`

Conversely, for the EFE-CCM we define the union operator Υ_i as Eq. 4.3 and Eq. 4.4 for the (PCA)FE-CCM and the (LDA)FE-CCM instantiations respectively.

$$\Phi_i(X) = \text{PCA}\left([\Psi_i(X), \hat{Y}_1^1, \hat{Y}_2^1], n\right) \tag{4.3}$$

$$\Phi_i(X) = \left[\text{PCA}\left([\Psi_i(X), \hat{Y}_1^1], n\right), \text{LDA}\left([\hat{Y}_2^1, Y_i^{gt,1}]\right)\right] \tag{4.4}$$

The biggest dissimilarity in the implementation of our work in comparison with that of Li et. al. in [Li 2012] relies in the number of tasks considered in the Holistic Scene Understanding formulation. While they include six different tasks for their experiments, namely Scene Categorization, Depth Estimation, Event Categorization Saliency Detection, Object Detection and Geometric Labeling, we restrict this work to the tasks of Scene Categorization and Object Recognition. In consideration of the main goal of this work, which is focused on the search of opportunities to improve cooperativity between Scene Classification and Object Recognition, we regard that this task simplification does not affect our sought goal. Additionally, as mentioned in previous chapters, the authors do not provide sufficient information to reliably replicate their work. Therefore, it is required for us to tune and redefine some important design parameters that directly affect the entire classification structure of the CCM (see Chapters 3.3.4, 4.1.1, 4.2.2). Due to the aforementioned facts, differences between the results reported here and those published in [Li 2012] are to be expected.

4.1 General Structure

4.1.1 Implemented Classifiers

Following the design parameters of Li et. al. [Li 2012] and Heitz et. al. [Heitz 2009], our first classification layer is composed of multiple RBF-Kernel SVM classifiers. Unlike Li et. al. however, whose second classification layer is composed uniquely of Logistic Classifiers, we utilize Random Forest classifiers (RF) instead. With the RFs, we aim to increment the classification power of the CCM instantiations while reducing possible degradation of the classification capacity due to possible non-descriptive, non-discriminative features aggregated to the model from the classifier responses of the first layer. In order to inspect the suitability of Random Forest for this task, we compare three CCM instantiations, in which the second classification layer is composed of Logistic Classifiers, Random Forests and RBF-Kernel SVMs, respectively. Our comparison strategy relies on the classification performance of the second layer instantiations on a unique validation data set. The validation set is composed of 20% of the original data set for each task, generated by a random sampling process from the training data set under the constraint that the label-wise sample distribution in both training and validation sets remains equal. Subsequently, we train the CCM on the new training set (consisting of the remaining 80% of the original training data set) and evaluate their performance

Table 4.1: Classification performance comparison of second classification layer instantiations composed of Logistic Classifiers, Random Forests and SVMs

Classifier	Scene Categorization	Object Detection				
		Car	Person	Horse	Cow	Mean
Logistic	68.42%	82.86%	83.09 %	81.25%	63.55%	77.69%
RF	**72.68%**	**88.57%**	**88.32%**	**84.48%**	**67.36%**	**82.18%**
SVM	71.99%	87.82%	86.56%	82.77%	65.52%	80.66%

Note: Best results are marked in bold letters.

Table 4.2: Summary - Hyperparameter selection for our Random Forest Classifiers

Hyperparameter	Value
`noTrees`	512[1]
`activeVarCount`	5% $*$ `noFeatures`
`maxDepth`	∞[2]
`minSampleCount`	1[2]

[1] As recommended by Leo Breiman in [Breiman 2001].
[2] The eventual overfitting caused by these parameters in each decision tree diminishes with the bagging strategy used on their responses.

in the validation set. The obtained results are shown in Table 4.1. From the obtained results we can observe, that the RF-based classification layer consistently outperforms the remaining instantiations. We select the same hyperparameter configuration for all our RF classifiers. Our hyperparameter selection is summarized in Table 4.2.

Another important fact worth highlighting is related to our implementation of the Support Vector Machines (SVMs), which are extensively used in our experiments. Due to the imbalanced nature of some of the utilized training data sets (see Chapter 4.2.2) and the reported sensibility of SVMs towards overfitting in such scenarios [Akbani 2004], we utilize a regularization strategy over the classification performance on the validation dataset to equilibrate the importance of the samples being misclassified, such as the one introduced by Huang and Shu-Xin in [Huang 2005b]. Consider the extreme case of a training and validation data set composed of 99% negative samples and 1% positive samples. In such a case, classifying all the samples as negative, regardless of the actual feature descriptor, would produce a CCR of 99%, which could mistakenly be interpreted as a good classification result. In order to cope with training on uneven sets, we modify the misclassification penalization of every sample with a weight inversely proportional to the ratio of the sample class in the data set [Huang 2005b]. Formally, for our example case, the misclassification

Table 4.3: Classification performance of the SVM with regularized and standard training schemas on the Object Recognition Task

Version	Car	Person	Horse	Cow	Mean
Negative Samples (Test)	25.055	48.483	3.202	436	-
Positive Samples (Test)	668	872	129	84	-
Standard Train.	93.74%	90.55%	71.66%	70%	81.49%
Regularized Train.	**98.97%**	**98.55%**	**94.91%**	**78.35%**	**92.70%**

Note: Best results are marked in bold letters.

penalization of a positive sample would be $1/0.01 = 100$, while the misclassification penalization of a negative sample would be $1/0.99 = 1.01$. Hence, the classifier would focus on classifying positive samples correctly. We corroborated the effect of the used regularization technique on our baselines for the Object Recognition task, obtaining consistent improvement over the unregularized version (Table 4.3).

4.1.2 Feedback Step: Implementation of the optimization problem

In order to solve the optimization problem formulated for the feedback step (Chapter 3.3.4), we utilize the Knitro Nonlinear Optimization Solver [p.41, fn.2]. Due to the fact that, in our formulation, we consider the general case of a Holistic Scene Understanding task, usually composed of several constituent tasks, whose ground-truth labels can either belong to a finite set of values $Y = y_i \mid i \in \{1, 2, .., I_{\max}\}$, e.g. Scene Classification, or to a continuous set of values $Y = y_i \mid i \in \Re^+$, e.g. Deep Estimation, we formulate our optimization problem as a Mixed-Integer Nonlinear Optimization Problem, also known as Mixed-Integer Nonlinear Programming (MINLP). In contrast to a NLP, a MINLP formulation allows us to force variables to exclusively take integral values during optimization. By doing so, our feedback step implementation is easily extendable to any possible task combination of Holistic Scene Understanding.

4.2 Baselines

In this section, we introduce the baselines utilized for the constituent tasks of our Holistic Scene Understanding approach. Note that the same baselines are used for CCM, FE-CCM and EFE-CCM.

4.2.1 Scene Categorization

For Scene Categorization, we utilize the MIT Outdoor Scene data set introduced by Oliva and Torralba in [Oliva 2001]. The scene classification task aims to classify an image into one of the eight categories: coast, forest, highway, inside city, mountain, open country, street and tall building. The input feature vector of the first layer

$\Psi_1(X) \in \Re^{512}$ is the GIST descriptor [Oliva 2001], extracted from 4×4 regions of the image on four scales and eight orientations. The output of the first layer scene classifier is $\hat{Y}_1^1 \in \Re^8$, an eight-dimensional vector containing the normalized odd-ratio of the image belonging to each category. We consider the MIT Outdoor Scene data set especially appropriate for our experiments, as multiple categories are very similar and allow for mistaken yet objectively correct shifting of the sample labels in the hidden layer. For example, some images might contain a coast scene with mountainous background. In such cases, one could objectively classify the sample correctly in both classes. We argue that this behavior permits us identify more abstract inter-task relationships between classes, e.g. correlation between the "horse" class and the agglomeration of natural scenes.

4.2.2 Object Recognition

For Object Recognition, we utilize the training and test set of PASCAL-VOC 2006 [Everingham 2006]. Although the data set contains 10 carefully labeled classes, namely bicycle, bus, car, motorbike, cat, cow, dog, horse, sheep and person, we exclusively consider the subset of classes car, person, horse and cow. Since the goal of this work is to demonstrate the framework ability to interchange information between tasks, we selected the object classes that interact the best with the converse tasks. For example, the correct detection of a car can already provide information towards the type of scene we are observing, specifically, towards human-made environments. On the other hand, the accurate detection of a cow or a horse would produce the same effect, this time towards natural scenes. Our object detection module builds on top of the part-based detector of Felzenszwalb et. al. [Felzenszwalb 2010]. Initially, we generate several candidate windows for each image by applying the part-based detector with a low threshold (overdetection). The input feature vector of the first layer $\Psi_2(X) \in \Re^{1765}$ is composed of the HOG features [Dalal 2005] extracted from each window candidate, plus the detection score from the part-based detector. The HOG features are extracted from the resized 64×64-pixel candidate window, with cell-size 8×8 pixels, block-size equal to twice the cell-size and an overlap equal to the cell-size during feature calculation. We consider unsigned gradients and utilize 9 bins for the histogram construction. There is an important restriction worth highlighting, imposed to the model caused by the utilization of the part-based object detector of Felzenszwalb et. al. [Felzenszwalb 2010]. Due to the fact that, following the design choices of the predecesor works of Li et. al. [Li 2012] and Heitz et. al. [Heitz 2009], we exclusively consider those candidate windows generated by the part-based object detector in the classification task, our overall detection capacity is restricted to that of the part-based detector. In other words, we cannot detect instances of objects, which have not been captured by the part-based detector. We aim, however, to improve the positive detection of those detected windows primarily classified as false positives.

The task of the first layer object classifier is to assign a 1 or 0 to each candidate window signalizing if the searched object is contained in the window or not. Following the standards of the PASCAL 2006 competition, we define a candidate window as correct if the Jaccard index $J(\hat{w}, w^{gt}) \geq 0.5$, where $\hat{w}$ and w^{gt} correspond to the candidate window and the ground-truth bounding box, respectively. In order to feed the first layer detection output to the second layer, we generate a detection map per image, where each candidate windows takes the value of the prediction (1 if positive, 0 if negative). We generate a separate detection map for every considered class in the object recognition task. Let $\hat{X}_c$ be the generated detection map of the c-th class and C be the total number of classes considered in the object detection task. The output of the object recognition module is defined as:

$$\hat{Y}_2^1 = [\hat{X}_c \; \forall \; c \in \{1, 2, ..., C\}] \tag{4.5}$$

One important design parameter that was not specifically defined by Li et. al. is the standardized size of the generated prediction maps. This parameter is crucial, since it directly affects the size of the feature vector utilized for the tasks in the second layer. Furthermore, as images can vary in their size, it is important to define a standardized prediction map size, to which all the generated feature maps are scaled. For the FE-CCM we utilize a prediction map size of 16×16 pixels, since we consider this dimension big enough to capture some of the spatial distribution of the objects in an image, without adding an extensive overhead in the complexity of the second layer tasks. The dimensionality of the additional generated features from the object tasks $\hat{Y}_2^1 = [\hat{X}_c \; \forall \; c \in \{1, 2, 3, 4\}]$ corresponds to $4 * 16^2$. For the first layer, we utilize C independent RBF-Kernel SVM classifiers, each of which aims to correctly classify the binary problem of false and true positive candidate windows generated for the c-th task by the part-based detector. Similarly, we include C binary RF Classifiers in the second classification layer.

4.3 Feature Reduction

In the Section 3.4, we introduced two instantiations of the EFE-CCM, namely the (PCA)FE-CCM and the (LDA)FE-CCM. In order to perform a reasonable comparison between instantiations, we implement them in such a way that the resulting vector Φ_i contains the same number of features in both cases. The input feature vector of the second layer in the FE-CCM can be calculated as $\dim \Phi_i(X) = \dim \Psi_i(X) + \sum_{i=1}^{n} \dim \hat{Y}_i^1$. For our particular use case, the dimension of the second layer input vectors is $\dim \Psi_1(X) = 512 + 8 + 4 * 256 = 1544$ and $\dim \Psi_2(X) = 1765 + 8 + 4 * 256 = 2.797$, respectively. For the (PCA)FE-CCM, we formulated the selection of principal components from the resulting PCA transformation as an optimization problem over the explained variance of the data set (Eq. 3.15). Here, we have one design parameter at disposal, which directly controls the number of resulting principal components, i.e. the threshold of explained variance $t_\Sigma \in \{0.0, ..., 1.0\}$. On the other hand, the (LDA)FE-CCM utilizes an identical

procedure as that of the (PCA)- instantiation, with the variation, that it is solely applied over the feature vector subset composed of $\Psi_i(X)$ and G, while an LDA procedure is implemented on the remaining feature set L. The proposed LDA feature reduction problem reduces the dimension of N generated detection maps into a single one. In our use case, this corresponds to a reduction of $4 * 256$ to 256 features. Since the LDA imposes a hard number of features, which need to be aggregated into the resulting feature vector, we select n in the (PCA)FE-CCM, such that the number of features in both instantiations are equal. In order to hold both instantiations as equal as possible, we solve Eq. 3.15 for $t_\Sigma = \{0.99, 0.999\}$ for both EFE-CCM instantiations and select the threshold t_Σ for which the difference of explained variance between them is minimal. A summary of the results is presented in Table 4.4. Further information can be found in the Appendix A.2, in which we plot the explained variance as a function of the number of principal components for all the classification tasks for both models. From the Table 4.4 we can see that the difference in the explained variance Δf_Σ is consistently lower for $t_\Sigma = 0.999$. It is worth noting, that we reach an average of 34% relative feature reduction, while roughly keeping the 99.9% of the data set variance in the original feature space.

Note that the resulting number of features after applying the feature reduction strategy over the OR_{horse} and the SC is remarkably lower than those obtained from the remaining tasks. One could erroneously interpret this as the corresponding data sets containing low variance in the original feature space, reason for which a high feature reduction is acquired. However, the real cause of this comes from other source: as the number of available samples in the data sets of OR_{horse} and the SC (720, 800 respectively) is lower than the number of features of the corresponding training data sets (2797, 1544), the singular value decomposition $S = U\Sigma V^*$ of the covariance matrix S generates a singular value matrix Σ, whose rank is determined by $\min\{n_f, n_s\}$, where n_f, n_s correspond to the number of features and samples in the dataset, respectively. As a result, the obtained PCA transformations are able to generate a maximum of 720 and 800 orthogonal linear combinations respectively and the solution of the Eq. 3.15 delivers an extremely low number of features. For further information about the presented problematic, we refer the reader to [Jolliffe 2016].

In order to generate a reasonable feature reduction on these data sets, we set the feature reduction ratio for these tasks to the average of the remaining tasks (34%) and the actual number of resulting features n to the minimum between the required number of features to achieve the imposed feature reduction ratio and the rank of the singular value matrix Σ of the covariance matrix S of the data set. Formally, we define n in these cases as:

$$\min_n \left[\left(1 - \frac{n}{N} = r\right), rank(\Sigma)\right] \tag{4.6}$$

Table 4.4: Summary - Difference in the explained Variance Between the (PCA)EFE-CCM and the (LDA)EFE-CCM for all the included tasks in the model.

Instantiation	OR_{car}	OR_{person}	OR_{horse}	OR_{cow}[1]	SC[1]
(LDA)99%[2]	1002	1114	1116	575	188
(PCA)99%	1052	1115	997	452	129
$f_\Sigma(n(\text{LDA}))$[3]	99.35	99.42	99.64	-	-
Δf_Σ	0.35	0.42	0.64		
(LDA)99.9%[2]	1567	1610	1581	693	342
(PCA)99.9%	1977	2014	1766	657	277
$f_\Sigma(n(\text{LDA}))$[3]	99.83	99.81	99.925	-	-
Δf_Σ	**-0.07**	**-0.09**	**0.025**	-	-
No. Features	1823	1866	1837	949	598
Reduction (%)	34.82	33.28	34.32	*66.07*[4]	*61.27*[4]

[1] We do not consider OR_{cow} or SC during the decision making process, as the number of training samples is smaller than the number of features or each tasks, reason for which a very low number of PCs is obtained.

[2] Note that the LDA instantiation becomes additional 256 from the LDA feature reduction part.

[3] $f_\Sigma(n(\text{LDA}))$ evaluates the resulting obtained variance when utilizing the total number of features from the corresponding (LDA)EFE-CCM instantiation.

[4] Provided for the sake of completeness. These low values are, however, exclusively obtained due to the small proportion of the dataset and are, therefore, not trustworthy.

Note: Best results are marked in bold letters.

Where N describes the number of features of the original space and r represents the imposed feature reduction ratio. For the OR_{horse} and the SC tasks, the solution of Eq. 4.6 delivers 720 and 800 features, respectively.

Chapter 5

Experiments and Results

Contents

In this chapter, we provide a thorough explanation of the experiments carried out in this work as well as our experimental settings and obtained results. Subsequently, we discuss our results, as well as the sources of experienced improvements and degradations.

5.1 Experimental Settings

We evaluate our proposed method over the combination of the tasks defined in Section 4 and compare its results with those obtained by the FE-CCM, the CCM and the baselines. For each constituent task, the performance assessment is carried out on the introduced standard data set of that subtask. The performance evaluation of the Scene Classification task ($i = 1$) is carried out by measuring the rate of incorrectly assigned labels on the test data set, while for the Object Recognition task ($i = 2$), we measure the average precision of the considered class labels. The total performance of a CCM instantiation (i.e. CCM, FE-CCM and EFE-CCM) is calculated as the average of the performance obtained for the constituent tasks. Note that originally, the CCM is exclusively utilized for homogeneous data sets [Heitz 2009]. Here, we adapt the evaluation procedure of the CCM to heterogeneous data sets utilizing the same evaluation scheme proposed for the FE-CCM and the EFE-CCM. Subsequently, we confront the execution times required for each of the CCM instantiations during inference.

In our experiments, both the EFE-CCM and the FE-CCM show convergence after 2 iterations. Regarding the CCM, one can attribute that its training procedure converges directly, as its training strategy does not present any iterative behavior.

5.2 Data Sets

Despite the fact that the data sets used in our experiments have been already introduced in the Chapter 4.2, there are some additional relevant details that need to be approached to completely define our experimental set-up. During our experiments, we utilize the same training/test set division as the baselines in their respective works. Specifically, for Scene Classification, we randomly select 100 images per class from the data set to create a training and test set with 800 and 1.888 images respectively. For the Object Detection task, the training and test sets have been previously defined in the PASCAL-VOC 2006 [Everingham 2006], containing 1.277 and 2.686 images respectively. As stated in Chapter 4.2.2, the Object Recognition Task is fractionated into C binary classification problems over several candidate windows, where C is the number of considered classes. The obtained number of candidate windows in the training and the test data set of each class are summarized in Table 5.1.

Table 5.1: Summary - Number of samples in the training and test data sets per class in the PASCAL-VOC 2006 dataset

Data Set	Car	Person	Horse	Cow
Training Set	25923	49555	3531	720
Test Set	26706	49948	3753	707

Note that, although we previously divide the respective training sets into training and validation sets for parameter tuning and selection (see Chapter 4.1.1), after the tuning process has been concluded, we reconstruct the original data set for subsequent tests. The results reported in this work are obtained from CCM instances trained on the complete training sets.

5.3 Results

The results obtained from our experiments regarding the classification accuracy are summarized in Table 5.3. Fig. 5.1 shows the confusion matrices for the baseline, the CCM and the FE-CCM for the scene classification task. Unfortunately, in contrast to the results reported by [Heitz 2009] and [Li 2012], our classification cascade deteriorates in comparison with the baselines for all the CCM instantiations. Due to this rather surprising conduct, we thoroughly analyze our structure in search for the sources that give raise to this behavior.

After corroborating our software solution, we concluded that the reason for this deterioration is strongly linked with the definition of the detection map size. During the feed-forward step, due to the fact that we utilize an standardized size for all the detection maps of 16×16 pixels, the detection maps seem not to capture enough of the spatial distribution of the objects and hampers our

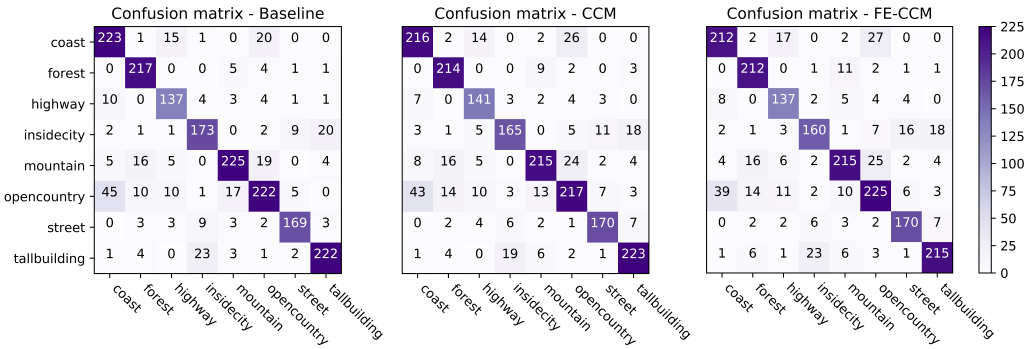
Table 5.2: Summary - Obtained Classification Accuracy Results

Model	Scene Categorization	Object Detection				
		Car	Person	Horse	Cow	Mean
Baselines[1]	**82.73%**	**98.98%***	**98.56%***	**94.96%***	**78.5%***	**92.75%***
CCM	82.68%	98.57%	98.33%	**94.48%**	77.37%	92.19%
(64)CCM	**82.89%***	**98.67%**	**98.37%**	94.40%	**77.95%**	**92.35%**
FE-CCM	81.89%	98.47%	98.28%	94.67%	75.53%	91.74%
(PCA)FE-CCM	75%	98%	98.20%	93.79%	75.39%	91.35%
(LDA)FE-CCM	**77.91%**	**98.23%**	**98.34%**	**94.06%**	**76.95%**	**91.89%**
Performance Decay[2,3]						
(PCA)FE-CCM	6.89%	0.47%	0.08%	0.88%	0.14%	0.39%
(LDA)FE-CCM	**3.98%**	**0.24%**	**-0.06%**	**0.61%**	**-1.42%**	**-0.15%**

[1] The baseline results correspond to the 1st classification layer responses.
[2] A negative decay corresponds to a classification improvement.
[3] The performance decay is calculated in comparison with the FE-CCM.
Note: Best results in each section are marked in bold letters.
Note: Best overall results are marked with (*).

efforts to enable communication between the constituent tasks. For this reason, the concatenated additional features seem to include noise in the model rather than provide additional information from the converse tasks, which simultaneously augments the Hughes-like effects in our second classification layer. Note, however, that due to the utilization of Random Forest, a classification strategy well known for its relative high robustness against the negative effects of "irrelevant" features (see the comparison with other classifiers in Table 4.1), an impressive damper of these negative effects is obtained. Furthermore, due to the fact that multiple instances disappear during the resizing procedure (Fig. A.1), we strongly believe that our incorporated feedback interface is not able to provide accurate information to the first layer about the actual totality of objects encountered in an image. Consequently, our efforts towards the optimum assignments of the first layer ground-truth labels are hampered and premature convergence is obtained.

In order to prove our hypothesis, we create a new CCM instance with a standardized detection map size of 64×64. Unfortunately, due to time restrictions, we are not able to prove the effect of this change for the FE-CCM structure. However, we are able to experience forceful results from this singular experiment, which influence the FE-CCM instances as well. Table 5.3 shows the results obtained for this CCM variant, referred to as (64)CCM. Interestingly, in contrast to all other CCM instantiations, we are able to see a slight improvement over the baselines for the scene classification task. Note, however, that improvement is just obtained for the scene categorization task, while for the complementary tasks, just a reduction in the performance diminishment is reached. We argue that this conduct is produced due to the fact that the scene categorization data set is composed of fixed-size



(a) (b) (c)

Figure 5.1: Confusion matrix of (a) the baseline, (b) the CCM and (c) the FE-CCM for the scene classification task.

images, while images in the object recognition data set strongly vary in size. As a result, images in the object recognition data set are stretched and skewed during the formation of our standardized squared detection map. Consequently, proper spatial reasoning on the images becomes more difficult. For instance, we strongly believe that a more sophisticated way of defining the standardized detection map is required. Note that none of the works we based our work on, provide information regarding the utilized detection map sizes nor any approach towards finding a good configuration.

A final fact that one must hold in mind is that the improvements of the (64)CCM over the CCM come in cost of $4 * 64^2 - 4 * 16^2 = 15360$ additional features for each of the second layer classifiers. In conclusion, the definition of the detection map sizes plays an outstanding role in the classification and time performance of CCM architectures. Therefore, it describes a very worthwhile objective for further research.

5.3.1 Evaluation of the feature reduction techniques in the EFE-CCM instances

As for any other feature reduction technique, a certain performance drop in the classification task in comparison with that on the complete feature set is to be expected. The lowest section of the Table 5.3 summarizes the classification decay obtained for the EFE-CCM instantiations. Interestingly, our proposed (LDA)FE-CCM outperforms the results obtained by the (PCA)FE-CCM. We consider this an especially relevant fact, as our proposed (LDA)FE-CCM allows for the simultaneous integration of numerous tasks in the cascaded classification model without increasing the complexity of the second classification layer. On the contrary, our (LDA)FE-CCM is able to use these additional "wavelength bands" incorporated in the model to improve the spatial LDA feature reduction. Table 5.3 summarizes the required

Table 5.3: Summary - Execution times of the CCM instantiations

Models	Duration(ms)[2]						
			1st Layer				
			SC	OR_{car}	OR_{person}	OR_{horse}	OR_{cow}
Baselines[1]	-	-	0.56	0.43	0.44	0.41	0.49
	Union		2nd Layer				
	SC	OR	SC	OR_{car}	OR_{person}	OR_{horse}	OR_{cow}
CCM	2.56	1.01	0.30	0.11	0.12	0.09	0.24
FE-CCM	2.56	1.01	0.30	0.11	0.12	0.09	0.24
(PCA)FE-CCM	1.81	0.65	0.25	0.08	0.09	0.08	0.19
(LDA)FE-CCM	**1.52**	**0.49**	**0.25**	**0.08**	**0.09**	**0.08**	**0.19**

The reported results correspond to the average processing time per sample. Note that an image include several samples for the OR task

[1] Note that the baseline times correspond to the first-layer execution times for all the CCM instantiations.

[2] The total execution time for the i-th task is calculated as the sum of the inference times for the classification tasks in the first layer plus the cost of the union operation and the execution time of the i-th task in the 2nd layer. In other words: Total $= \sum_{1^{st}\ \text{layer}} C + \sum_{\text{union}} C + C_i$.

Note: Best results are marked in bold letters.

execution times for the CCM instantiations and the baselines. As expected, the EFE-CCM instances perform remarkably faster than the FE-CCM and the CCM structures. This velocity improvement comes in cost of an average performance decay of $[6.89\%, 0.39\%]$ and $[3.98\%, -0.15\%]$ for the (PCA)- and the (LDA)FE-CCM instantiations in the scene classification and object recognition tasks, respectively. Interestingly, the (LDA)FE-CCM is able to generate slight improvements over the FE-CCM for some of the constituent subtasks of the object recognition task. This behavior is caused due to the elimination of non-descriptive features in the detection maps and the beholding of spatial information achieved by our proposed spatial LDA reduction scheme.

5.4 Effect of the first layer response

One interesting advantage of utilizing Random Forest Classifiers for the second layer classification layer, is the possibility to easily calculate the importance of each variable for the classification task. In OpenCV, this is defined by setting `setCalculateVarImportance(true)`. The variable importance provides us with a measure of the relevance of independent features for the obtained accuracy. For the scope of our work, we can utilize this measurement to estimate the influence that each of the constituent sub-groups of our input feature vector $[\Psi_i(X), \hat{Y}^1, \hat{Y}^2]$ has

in the classification task. It is important to note that the training time required when calculating the variable importances is, on average, equal to 12× the average training time required without this setting.

5.4.1 Scene Classification

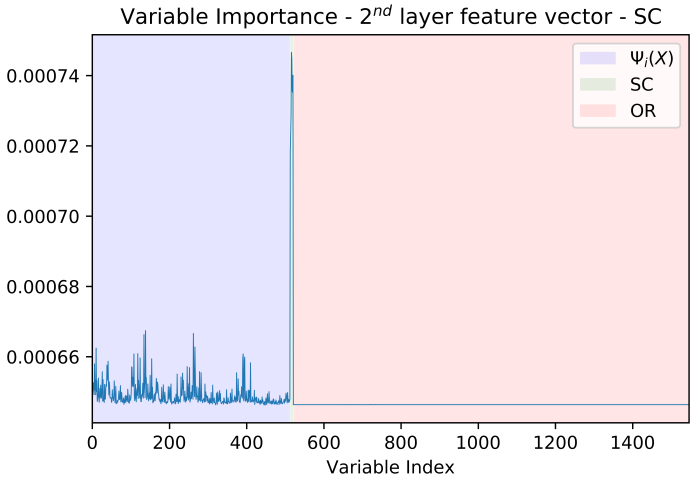


Figure 5.2: Variable Importance for the Scene Classification Task

Fig. 5.2 shows the importance of each feature vector component for the scene classification task. Interestingly, the variables with the highest importance are those corresponding to the response of the first-layer scene classifier $\hat{Y}^1$. Note that its relative importance is remarkably higher than that of the surrounding features (Fig. 5.3a). On the other hand, the output responses of the object recognition task, seem to have a rather moderate importance in the task. In order to take the discussion about the effect of the first layer responses further, we evaluate the importance of each sub-group separately. Fig. 5.3a shows the relative importance of each class in the scene categorization, while Fig. 5.3b shows the reconstructed 2-dimensional importance maps corresponding to the detection maps of each class in the object recognition task.

Note that the lower half of the car detection maps exhibits some peaks in the importance maps. Interestingly, this behavior matches the usual spatial distribution of cars in outdoor images. Based on this result, we can conclude that the second layer is able (to some extend) to capture contextual information provided by the object recognition task. Conversely, the importance of the detection maps for the remaining classes is uniform across the entire image. Subject to the fact that the scene classification data set strongly focuses on panoramic views, we argue that the restricted capacity of the algorithm to detect additional object relationships in the scene classification data set is mainly caused by two reasons: firstly, in

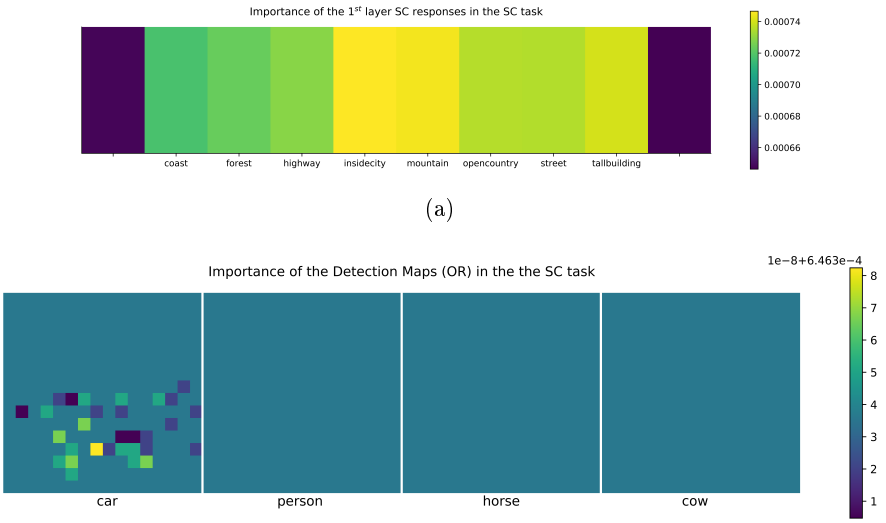


Figure 5.3: Variable Importance Maps of the 1st Layer (a) Scene Classification and (b) Object Detection outputs for the Scene Categorization task

contrast to cars, which prominently appear in specific areas across panoramic images, animals and persons do not follow any distinctive spatial ordering. As an effect, the algorithm is not able to prioritize any spatial distribution for these classes. Secondly, we argue that these uniform maps are also caused due to the resizing process utilized during the construction of the detection maps. The scene classification data set is composed uniquely by panoramic 256×256-pixel sized images, while our detection map dimensions are set to 16×16 pixels. Consequently, any positive detection on regions, whose dimension is minor than 16 pixels in any direction, disappears during the resizing procedure. We strongly believe that as a consequence, a large amount of contextual information is lost during this procedure, which simultaneously, hampers the capture of richer spatial distributions for classes, whose instances are usually detected in small regions. In Fig. A.1, we provide an example in which several instances disappear during the resizing step.

Note that cars usually cover a large part of outdoor images in human-made scenarios, reason for which our algorithm is able to detect a rough spatial importance prioritization after the resizing process.

5.4.2 Object Recognition

There is an essential difference between the scene classification data set and the one utilized for object recognition. Due to nature of the classification task, the images in the object recognition data set focus in capturing object instances in various poses, while the scene categorization data set aims to capture the composition of an entire scene (Fig. 5.4). Consequently, the object instances in the object recognition data set usually cover a substantially larger area in the image, reason for which the detection maps play a more interesting role in the importance distribution.

(a) (b)

Figure 5.4: Exemplary images from the Scene Classification data set (a) and the Object Recognition data set (b)

Fig. 5.5 shows the distribution of the variable importance for each binary constituent classification task in the object recognition assignment. Note that, in contrast to the scene classification task, the detection map of the first layer for the same task, together with the 1765^{th} feature of the original feature vector $\Psi_1(X)$, reveal the highest importance in the task. Recall that the 1765^{th} feature in $\Psi_1(X)$ corresponds to the detection score of the part-based object detector of Felzenszwalb [Felzenszwalb 2010]. Interestingly, the first layer scene classification responses exhibit a rather moderate importance for the task. Fig. 5.8 and Fig. 5.6 show the importance distribution of the outputs of the first layer scene classification and object recognition tasks, respectively.

Although the detection maps exhibit a more active behavior than in the scene classification task, at first glance, they do not provide much additional information regarding any interchange of contextual information between the classes involved in the object recognition task. Note that the peaks in the importance maps appear exclusively in the detection map corresponding to the same class. Interestingly, these peaks are mostly gathered in the center of the image. The reason for the later conduct is that the photographies included in this data set are, to a great extent, focused in capturing instances of the objects, whose classes we want to recognize. Consequently, these photographies are usually centered towards the corresponding

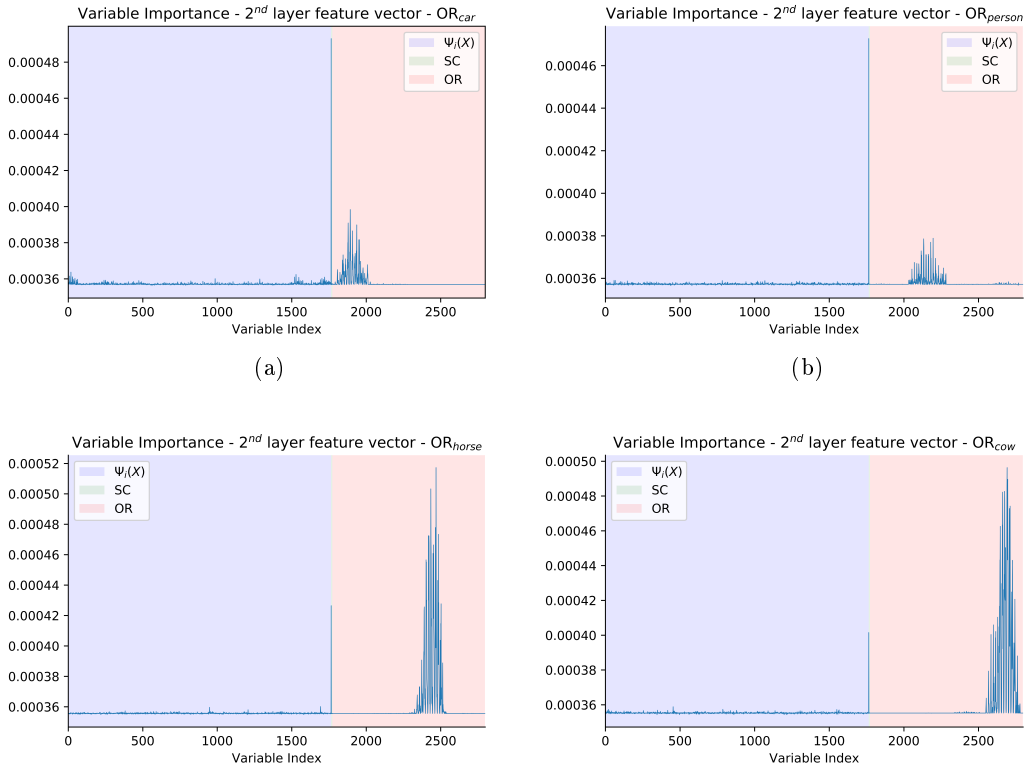


Figure 5.5: Variable Importance Distribution of the (a) car, (b) person, (c) horse, (d) cow recognition task.

objects (Fig. 5.4b). Despite the fact that Fig. 5.6 does not show any spatial importance distribution for other than the beheld task, the algorithm does capture (to some extend) spatial relationships between the constituent object detection tasks (Fig. 5.7). The reason for which we are not able to appreciate these relations in Fig. 5.6 is due to the fact that the very strong importance peaks in the detection maps of the regarded task undermine the more subtle dependencies with other classes. With an adequate local scale for each detection map, however, we are able to recognize some interesting spatial relationships between classes. Note that the importance maps of the scene classification task (Fig. 5.3b) exhibit the exact same uniform behavior when utilizing this local-based scaling in their graphical representation.

Interestingly, the importance maps of Fig. 5.7 reveal very intuitive spatial relationships between objects. Furthermore, these relationships are consistent across the binary constituent problems. For instance, the spatial relationships of the class cow in the car classification problem are consistent with those exhibited by the class car in the cow classification task. This conduct provides us with a verification tool for the veracity of the algorithm.

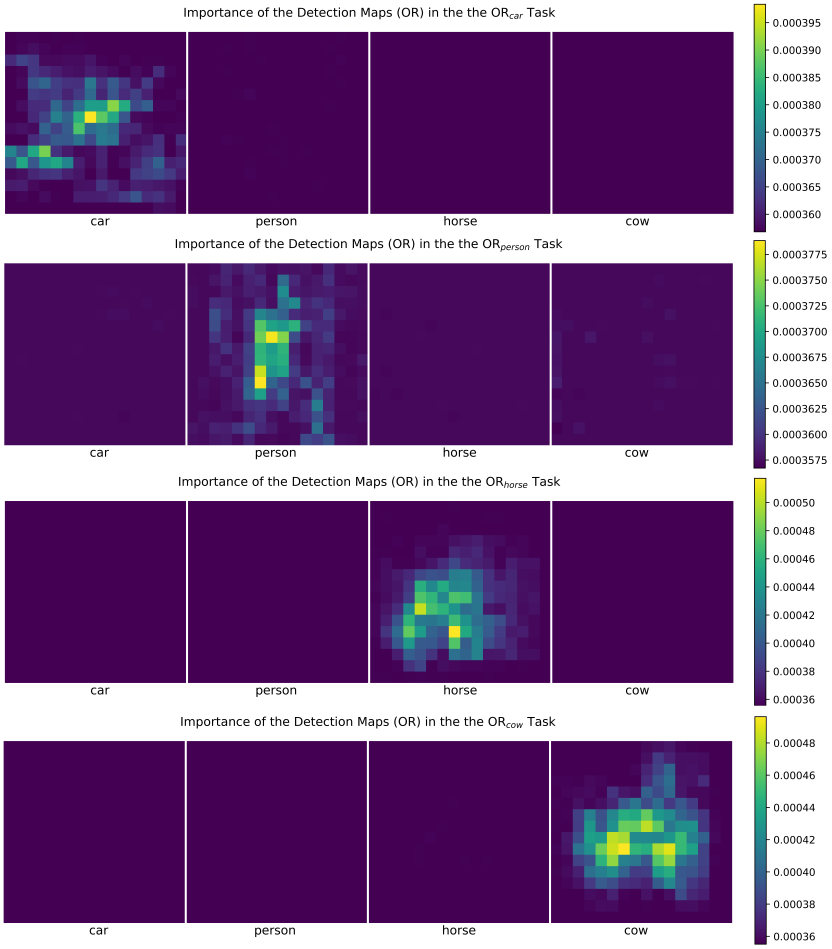


Figure 5.6: Importance maps for the 1st layer object recognition responses.

Note that the importance distribution follows a rough form of the considered object class. For instance, the spatial importance distribution of the classes car, horse and cow follow a horizontal distribution, while the class person is related to a rather narrow tall structure. Furthermore, the object correlations detected by our algorithm are commonly observed throughout the data set. Exemplarily, the importance maps of the class car describe that cars often appear simultaneously with humans but not much with horses or cows. Persons interact with all the other classes. Horses intensively interact with persons and cows but less with cars, and cows often appear together with horses but much less with humans and seldom with cars.

Complementarily, these importance maps do not exclusively provide us iformation about appearance correlations, but as a matter of fact describe likely relative positioning between the object instances. Exemplarily, cars often appear in the upper right corner of images focused on persons. This is a very common and

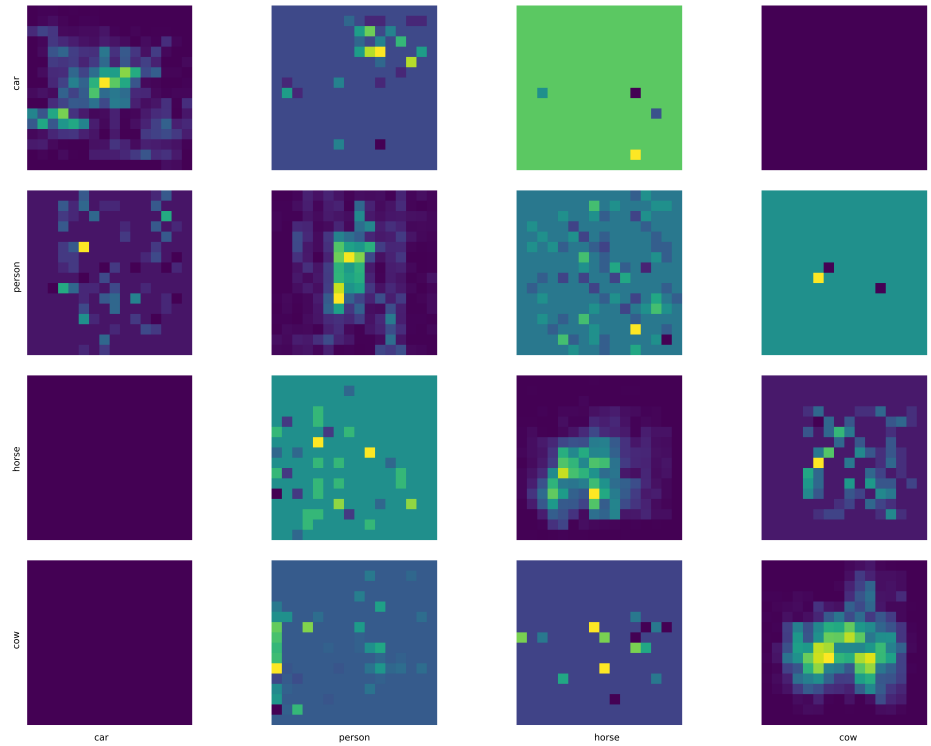


Figure 5.7: Detailed importance maps with local-based scaling for the 1st layer object recognition responses of the (a) car, (b) person, (c) horse, (d) cow recognition task.

descriptive situation for pictures in inside city scenes, in which cars play a decisive function as background (Fig. A.4). For images of horses, humans are likely to appear in any desired position surrounding the horses, either mounting them, or in front and at their side while posing or feeding them. On the other hand, cows do not appear on horses but much likely in their lateral vicinities. This conduct is very well represented in Fig. 5.7c. In short, our algorithm does not only become aware of object correlations to benefit the classification task but of their relative positioning too.

It is exciting to see, that our algorithm is able to capture inter-task relationships between the object recognition and the scene categorization as well (Fig. 5.8). For instance, cars usually appear in human-made sceneries, in which the probabilities of encountering roads is high. Our method is able to detect this relationship and assigns a high importance towards the classes highway, street and tall-building and very low ones towards classes coast or forest, where cars are very unlikely to appear. Persons are likely to appear across any type of scene. Horses typically appear in scenes with a large area covered by trees (forest- and mountain-like) or in scenes containing roads or streets (Fig. A.3). To finalize, the algorithm relates the appearance of cows with mountain-like scenes. This is interesting, as one

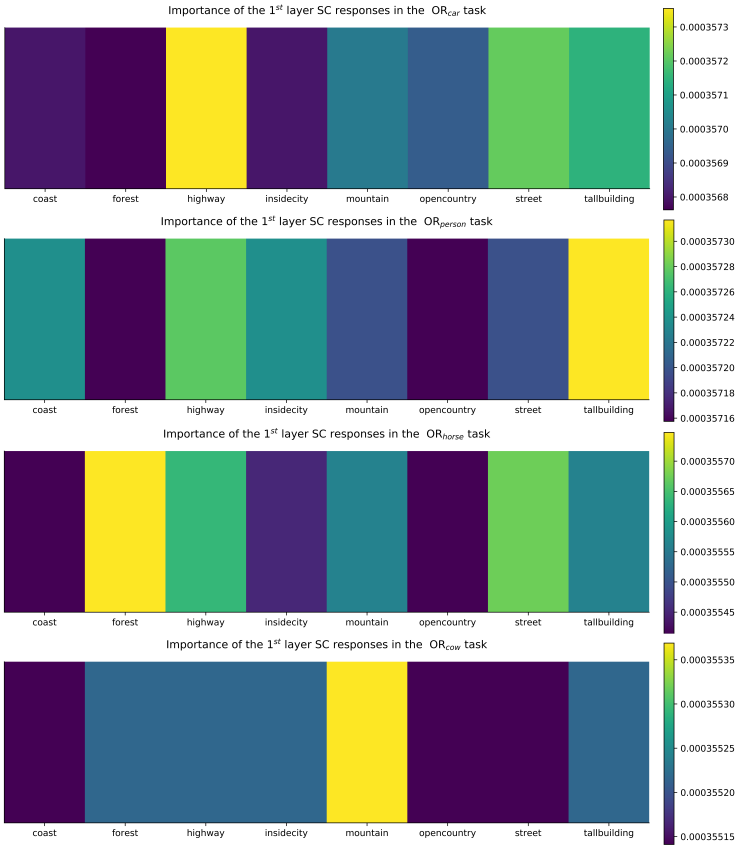


Figure 5.8: Variable Importance for the 1st layer scene classification responses.

would expect cows to appear more often in open country-like scenes. However, we perceived that images from the open country class in the scene classification data set oftentimes include large sky areas in the picture. As the images in the object recognition data set focus in the object itself, they usually include large grass-like areas with less to none sky appearance (Fig. A.2). Due to this reason, it is intuitive to obtain a stronger relationship towards mountain-like scenes.

Despite the fact that our proposed approaches do not achieve any remarkable improvement in comparison with the baselines, we strongly believe that further research into cascaded classification models, particularly into our (LDA)FE-CCM, could lead to a strong, flexible and large-scale adequate framework for joint optimization of complex classification structures. We consider that the capacity of our algorithm to grasp inter-task and inter-object spatial relationships is worth examining in further research. In Section 6.2, we provide various future work directions we believe could lead to large improvements of the results obtained in this work.

Chapter 6

Conclusions and Future Work

Contents

6.1 Conclusions

In this work we introduced the Enhanced Feedback Enabled Cascaded Classification Models (FE-CCM) method, an extension of the previous FE-CCM and CCM strategies, which aims to strengthen some of the most important deficiencies of its predecessor works, while upholding all of their advantages.

Our method allows for the seamless combination of state-of-the-art classifiers into a unified framework, which is able to steer the classifiers towards an optimum. In order to do so, our algorithm exclusively requires a "black-box" interface for each classifier and thus, allows for joint optimization without requiring any modification of the inner working structure of the constituent classifiers. Our method includes a feedback interface in the training algorithm, which permits deeper classifiers to encourage shallower ones to focus on correctly classifying relevant mistakes that negatively impact the classification performance "downstream". Additionally, we include an inter-layered feature reduction approach, which compresses the feature vector of the second classification layer, composed of the original features of each task and the (correlated) classifier responses from the first layer, into a low-dimensional vector, which beholds (most of) its discriminant power. We provide two instantiations for the EFE-CCM, the (PCA)FE-CCM and the (LDA)FE-CCM, which differ from each other in the working of the inter-layered feature reduction scheme. While (PCA)FE-CCM utilizes an unsupervised strategy for the feature reduction, (LDA)FE-CCM combines supervised and unsupervised feature reduction techniques to generate a more descriptive low-dimensional vector, which considers the actual label-wise distribution and beholds meaningful spatial information.

It is important to highlight that this framework is very general and can be easily applied across several machine learning fields to seamlessly combine complex classifiers. Furthermore, our proposed framework is able to include a large number of tasks into a single framework, without producing a large complexity overhead over the utilized baselines. Furthermore, our algorithm permits the inclusion of disjoint data sets in the model, which contain ground-information for a single task. This later property increases the suitability of our algorithm for practical applications.

In our experiments for simultaneous scene classification and object recognition, we show that our unified model provides means of communication between the constituent tasks, while imposing a low overhead for the inference task in comparison with predecessor CCM instantiations. Despite the fact that our model is not able to improve the classification performance of the baselines, we strongly believe that further research into the cascaded classification models, particularly into our (LDA)FE-CCM, could lead to a strong, flexible and large-scale adequate framework for joint optimization of complex classification structures.

At the completion of this work, we leave various open questions worthwhile for further research. We strongly believe that their solution will lead to improvements in several aspects of this work and extend both its performance and its applicability. In Chapter 6.2 we summarize some interesting directions for further research and provide first approaches towards their solution.

6.2 Future Work

Due to time restrictions, it was not possible for us to test a larger scope of variations and configurations for our proposed framework. In this chapter we summarize some complementary ideas we believe could lead to large improvements.

6.2.1 Inclusion of additional tasks and reformulation of the inter-layered unsupervised feature reduction scheme

As mentioned in Chapter 1, the most straight-forward way to improve the overall performance of the CCM instantiations is to include additional tasks in the model [Li 2012]. However, due to the Hughes-like effects, the unbalanced incorporation of additional features in the second layer classifiers would eventually lead to a performance diminishment. Thanks to the included inter-layered feature reduction procedure in the EFE-CCM, we are able to reduce these negative effects. Unfortunately, however, as the number of features drastically increases without any additional samples, the estimation of the feature reduction parameters deteriorates too. Therefore, it is strongly recommended to include additional samples in the feature reduction formulation.

In the case of an unsupervised feature reduction scheme, one possible alleviation to this problematic would be the inclusion of the entire data set available in the first classification layer in the feature reduction formulation. Consequently, as the number of tasks increases, so do the number of available samples. Although these additional samples could lead to a better estimation of the feature reduction parameters and, with it, to a better feature reduction, there is an important fact one needs to be aware of. As unsupervised feature reduction does not consider the label-wise distribution of the samples but rather takes usage of statistical properties of the data set, the inclusion of additional samples that do not follow the statistical properties of the original data set could deviate our approach from its sought goal. As an illustration, assume a holistic scene understanding approach composed of the same scene categorization task as the one handled in this work and other task constituted mainly by outer space images. As there would likely not be major statistical relationships between the components of this task and we do not consider the labels of each task, the merged data set would likely exhibit a different statistical behavior than the data set of the scene classification task alone. In other words, the feature reduction could be misled.

6.2.2 Reformulation of the detection map content

Following the design parameters of Li et. al. [Li 2012], we utilize binary thresholding on the response of the first layer object recognizer for the construction of the detection maps (see Chapter 4.2.2). As a result, the generated detection maps are fully composed of positive (1) and negative (0) regions. Since a detection map containing the actual response probabilities for each region holds richer contextual information, e.g. regions with probability values in range $[0.0, 0.49]$ instead of regions solely formed by zeros, we believe that this variation in the detection map construction could be advantageous for the discrimination power of second layer classifiers. For example, the classifiers could make usage of the object's appearance uncertainty across the detection map to produce more reliable inference.

Likewise, we argue that such detection maps are very likely to improve our inter-layered feature reduction as well, both for unsupervised and supervised strategies. For instance, we believe that the (PCA)FE-CCM could strongly benefit from the enhanced variance obtained with the inclusion of such a detection maps in the estimation of the principal components. Likewise, our (LDA)FE-CCM would also be able to produce a more precise feature reduction. In our study case, in which we solely dispose of four "wavelength bands", it is intuitive to think that the binarization procedure imposed in the predecessor CCM instances reduce the description power of our resulting unified detection map. We are certain that the utilization of the actual probability maps would provide a greater extent of information, which would allow for our LDA to find a more powerful feature space transformation, than that obtainable from a $\{0,1\}^4 \rightarrow \Re$ scenario.

6.2.3 Formulation of the optimal detection map size

The works of [Heitz 2009] and [Li 2012] leave one very relevant aspect in the creation of the classification cascade unattended, which has a very distinctive and remarkable effect on the model performance and complexity: the standardized detection map size. In our results (Chapter 5.3), we show that this parameter has an strong effect on the overall performance of the model and is one of the main reasons of the performance decay of our work in comparison with the results reported in [Heitz 2009] and [Li 2012]. We argue that a formulation towards the optimal detection map size is indispensable for the aim of the CCMs to construct a unique, reliable and general-purpose unifying framework for joint optimization. Moreover, such a formulation should consider the inclusion of data sets, in which the images strongly vary in their dimensions, without causing any strong skewing or stretching to allow for proper spatial reasoning.

In order to avoid skewing and stretching during the detection map construction, one could include padding in the images to construct an standardized image size per data set. One feasible option could be to add a constant black padding to the centrally aligned image, until achieving the standardized size. Consider W_s and H_s the initial standardized width and height of a data set D. Additionally, consider r_s^* the standardized image size ratio ($\frac{H}{W}$) of D. We can define W_s, H_s and r_s^* as the respective mode values in the data set D, i.e. $\{W_s, H_s\} = \text{Mo}(\{W_i, H_i\}\ \forall i \in D)$ and $r_s^* = \text{Mo}(r_i\ \forall i \in D)$. Subsequently, we can define the standardized image size W_s^*, H_s^* of a data set as the next following combination of W_s, H_s such that r_s^* holds.

With the aforementioned formulation, we produce a one-dimensional optimization problem for the selection of the detection map size over the width of the detection map, subject to the restriction that r_s^* is kept constant. Subsequently, one could define a validation data set or utilize cross-validation to find the detection map width that produces the highest classification rate (e.g. by means of a grid-search). One eventual problem to bear in mind is that such a training scheme could possibly lead to extremely long training times, which could reduce the applicability of the detection map formulation for practical situations.

Bibliography

[Abbeel 2007] Pieter Abbeel, Adam Coates, Morgan Quigley and Andrew Y Ng. *An application of reinforcement learning to aerobatic helicopter flight.* In Advances in neural information processing systems, pages 1–8, 2007. 9

[Akbani 2004] Rehan Akbani, Stephen Kwek and Nathalie Japkowicz. *Applying support vector machines to imbalanced datasets.* In European conference on machine learning, pages 39–50. Springer, 2004. 43

[Anguelov 2005] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers and James Davis. *SCAPE: shape completion and animation of people.* In ACM transactions on graphics (TOG), volume 24, pages 408–416. ACM, 2005. 9

[Bappy 2016] Jawadul Hasan Bappy and Amit K Roy-Chowdhury. *Inter-dependent CNNs for joint scene and object recognition.* In Pattern Recognition (ICPR), 2016 23rd International Conference on, pages 3386–3391. IEEE, 2016. 13

[Baudat 2000] Gaston Baudat and Fatiha Anouar. *Generalized discriminant analysis using a kernel approach.* Neural computation, vol. 12, no. 10, pages 2385–2404, 2000. 16

[Bengio 2007] Yoshua Bengio, Yann LeCun*et al. Scaling learning algorithms towards AI.* Large-scale kernel machines, vol. 34, no. 5, pages 1–41, 2007. 10

[Bouman 1994] Charles A Bouman and Michael Shapiro. *A multiscale random field model for Bayesian image segmentation.* IEEE Transactions on image processing, vol. 3, no. 2, pages 162–177, 1994. 9

[Breiman 2001] Leo Breiman. *Random forests.* Machine learning, vol. 45, no. 1, pages 5–32, 2001. 43

[Buades 2005] Antoni Buades, Bartomeu Coll and J-M Morel. *A non-local algorithm for image denoising.* In Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, volume 2, pages 60–65. IEEE, 2005. 9

[Caruana 1998] Rich Caruana. *Multitask learning.* In Learning to learn, pages 95–133. Springer, 1998. 10

[Choi 2010] Myung Jin Choi, Joseph J Lim, Antonio Torralba and Alan S Willsky. *Exploiting hierarchical context on a large database of object categories.* 2010. 11

[Cohen 1991] Fernand S. Cohen, Zhigang Fan and Maqbool A Patel. *Classification of rotated and scaled textured images using Gaussian Markov random field models.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 13, no. 2, pages 192–202, 1991. 9

[Collobert 2008] Ronan Collobert and Jason Weston. *A unified architecture for natural language processing: Deep neural networks with multitask learning.* In Proceedings of the 25th international conference on Machine learning, pages 160–167. ACM, 2008. 7, 12

[Dalal 2005] Navneet Dalal and Bill Triggs. *Histograms of oriented gradients for human detection.* In Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, volume 1, pages 886–893. IEEE, 2005. 45

[Deng 2005] Huawu Deng and David A Clausi. *Unsupervised segmentation of synthetic aperture radar sea ice imagery using a novel Markov random field model.* IEEE Transactions on Geoscience and Remote Sensing, vol. 43, no. 3, pages 528–538, 2005. 9

[Deng 2009] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li and Li Fei-Fei. *Imagenet: A large-scale hierarchical image database.* In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, pages 248–255. Ieee, 2009. 14

[Deng 2011] Jia Deng, Alexander C Berg and Li Fei-Fei. *Hierarchical semantic indexing for large scale image retrieval.* 2011. 11

[Dieckmann 1997] Ulrich Dieckmann, Peter Plankensteiner and Thomas Wagner. *SESAM: A biometric person identification system using sensor fusion.* Pattern recognition letters, vol. 18, no. 9, pages 827–833, 1997. 8

[Dong 2007] Ming Dong and David He. *Hidden semi-Markov model-based methodology for multi-sensor equipment health diagnosis and prognosis.* European Journal of Operational Research, vol. 178, no. 3, pages 858–878, 2007. 8

[Durand 2015] Thibaut Durand, Nicolas Thome and Matthieu Cord. *Mantra: Minimum maximum latent structural svm for image classification and ranking.* In Proceedings of the IEEE International Conference on Computer Vision, pages 2713–2721, 2015. 9

[El-Bakry 2001] Hazem M El-Bakry. *Automatic human face recognition using modular neural networks.* Machine Graphics and Vision, vol. 10, no. 1, pages 47–73, 2001. 7

[Everingham 2006] Mark Everingham, Andrew Zisserman, Christopher Williams and Luc Van Gool. *The pascal visual object classes challenge 2006 (voc 2006) results.* 2006. 45, 50

[Faundez-Zanuy 2005] Marcos Faundez-Zanuy. *Data fusion in biometrics.* IEEE Aerospace and Electronic Systems Magazine, vol. 20, no. 1, pages 34–38, 2005. 8

[Fauvel 2013] M. Fauvel, Y. Tarabalka, J. A. Benediktsson, J. Chanussot and J. C. Tilton. *Advances in Spectral-Spatial Classification of Hyperspectral Images.* Proceedings of the IEEE, vol. 101, no. 3, pages 652–675, March 2013. 17

[Felzenszwalb 2010] P. F. Felzenszwalb, R. B. Girshick, D. McAllester and D. Ramanan. *Object Detection with Discriminatively Trained Part-Based Models.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 9, pages 1627–1645, Sept 2010. 45, 56

[Fink 2004] Michael Fink and Pietro Perona. *Mutual boosting for contextual inference.* In Advances in neural information processing systems, pages 1515–1522, 2004. 7

[Freund 1997] Yoav Freund and Robert E Schapire. *A decision-theoretic generalization of on-line learning and an application to boosting.* Journal of computer and system sciences, vol. 55, no. 1, pages 119–139, 1997. 7

[Friedman 1996] Jerome H Friedman. *Another approach to polychotomous classification.* Technical Report, Statistics Department, Stanford University, 1996. 30

[Galleguillos 2010] Carolina Galleguillos and Serge Belongie. *Context based object categorization: A critical survey.* Computer vision and image understanding, vol. 114, no. 6, pages 712–722, 2010. 10

[Gonfaus 2010] Josep M Gonfaus, Xavier Boix, Joost Van de Weijer, Andrew D Bagdanov, Joan Serrat and Jordi Gonzalez. *Harmony potentials for joint classification and segmentation.* In Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, pages 3280–3287. IEEE, 2010. 10

[Gould 2009] Stephen Gould, Tianshi Gao and Daphne Koller. *Region-based segmentation and object detection.* In Advances in neural information processing systems, pages 655–663, 2009. 12

[Graves 2009] Alex Graves and Jürgen Schmidhuber. *Offline handwriting recognition with multidimensional recurrent neural networks.* In Advances in neural information processing systems, pages 545–552, 2009. 7

[Gupta 2009] Abhinav Gupta, Aniruddha Kembhavi and Larry S Davis. *Observing human-object interactions: Using spatial and functional compatibility for recognition.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 10, pages 1775–1789, 2009. 11

[Gupta 2011] Abhinav Gupta, Scott Satkin, Alexei A Efros and Martial Hebert. *From 3d scene geometry to human workspace.* In Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, pages 1961–1968. IEEE, 2011. 11

[Guyon 2003] Isabelle Guyon and André Elisseeff. *An introduction to variable and feature selection.* Journal of machine learning research, vol. 3, no. Mar, pages 1157–1182, 2003. 14, 15

[Hansen 1990] Lars Kai Hansen and Peter Salamon. *Neural network ensembles.* IEEE transactions on pattern analysis and machine intelligence, vol. 12, no. 10, pages 993–1001, 1990. 7

[Hastie 1989] Trevor Hastie and Werner Stuetzle. *Principal curves.* Journal of the American Statistical Association, vol. 84, no. 406, pages 502–516, 1989. 15

[Heckerman 1995] David Heckerman, Dan Geiger and David M Chickering. *Learning Bayesian networks: The combination of knowledge and statistical data.* Machine learning, vol. 20, no. 3, pages 197–243, 1995. 8

[Heitz 2009] Geremy Heitz, Stephen Gould, Ashutosh Saxena and Daphne Koller. *Cascaded Classification Models: Combining Models for Holistic Scene Understanding.* In D. Koller, D. Schuurmans, Y. Bengio and L. Bottou, editeurs, Advances in Neural Information Processing Systems 21, pages 641–648. Curran Associates, Inc., 2009. 1, 2, 4, 8, 13, 19, 22, 23, 25, 26, 27, 34, 35, 42, 45, 49, 50, 64

[Hinton 2006] Geoffrey E Hinton, Simon Osindero and Yee-Whye Teh. *A fast learning algorithm for deep belief nets.* Neural computation, vol. 18, no. 7, pages 1527–1554, 2006. 8

[Hoiem 2008a] Derek Hoiem, Alexei A Efros and Martial Hebert. *Closing the loop in scene interpretation.* In Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, pages 1–8. IEEE, 2008. 13

[Hoiem 2008b] Derek Hoiem, Alexei A Efros and Martial Hebert. *Putting objects in perspective.* International Journal of Computer Vision, vol. 80, no. 1, pages 3–15, 2008. 10, 12

[Huang 2005a] Chang Huang, Haizhou Ai, Yuan Li and Shihong Lao. *Vector boosting for rotation invariant multi-view face detection.* In Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on, volume 1, pages 446–453. IEEE, 2005. 7

[Huang 2005b] Yi-Min Huang and Shu-Xin Du. *Weighted support vector machine for classification with uneven training class sizes.* In Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on, volume 7, pages 4365–4369. IEEE, 2005. 43

[Hughes 1968] Gordon Hughes. *On the mean accuracy of statistical pattern recognizers.* IEEE transactions on information theory, vol. 14, no. 1, pages 55–63, 1968. 14

[Imani 2015] Maryam Imani and Hassan Ghassemian. *Feature space discriminant analysis for hyperspectral data feature reduction.* ISPRS Journal of Photogrammetry and Remote Sensing, vol. 102, pages 1–13, 2015. 14, 16

[Izenman 2013] Alan Julian Izenman. *Linear discriminant analysis.* In Modern multivariate statistical techniques, pages 237–280. Springer, 2013. 16

[Jia 2013] X. Jia, B. Kuo and M. M. Crawford. *Feature Mining for Hyperspectral Image Classification.* Proceedings of the IEEE, vol. 101, no. 3, pages 676–697, March 2013. 17

[Jing-Xia 2015] C. Jing-Xia, Z. Yan-Ning, J. Dong-Mei, L. Fei and X. Jia. *Multi-class Object Recognition and Segmentation Based on Multi-feature Fusion Modeling.* In 2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom), pages 336–339, Aug 2015. 12

[Jolliffe 2016] Ian T Jolliffe and Jorge Cadima. *Principal component analysis: a review and recent developments.* Phil. Trans. R. Soc. A, vol. 374, no. 2065, page 20150202, 2016. 47

[Kennedy 2011] James Kennedy. *Particle swarm optimization.* In Encyclopedia of machine learning, pages 760–766. Springer, 2011. 8

[Klecka 1980] William R Klecka and William R Klecka. Discriminant analysis, volume 19. Sage, 1980. 16

[Koller 2009] Daphne Koller and Nir Friedman. Probabilistic graphical models: Principles and techniques - adaptive computation and machine learning. The MIT Press, 2009. 9

[Krizhevsky 2012] Alex Krizhevsky, Ilya Sutskever and Geoffrey E Hinton. *Imagenet classification with deep convolutional neural networks.* In Advances in neural information processing systems, pages 1097–1105, 2012. 7

[Kumar 2005] Sanjiv Kumar and Martial Hebert. *A hierarchical field framework for unified context-based classification.* In Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on, volume 2, pages 1284–1291. IEEE, 2005. 12

[Kuo 2004] Bor-Chen Kuo and David A Landgrebe. *Nonparametric weighted feature extraction for classification.* IEEE Transactions on Geoscience and Remote Sensing, vol. 42, no. 5, pages 1096–1105, 2004. 16

[Kuo 2009] Bor-Chen Kuo, Cheng-Hsuan Li and Jinn-Min Yang. *Kernel nonparametric weighted feature extraction for hyperspectral image classification.* IEEE Transactions on Geoscience and Remote Sensing, vol. 47, no. 4, pages 1139–1155, 2009. 16

[Lachenbruch 1979] Peter A Lachenbruch and M Goldstein. *Discriminant analysis.* Biometrics, pages 69–85, 1979. 16

[LeBlanc 1994] Michael LeBlanc and Robert Tibshirani. *Adaptive principal surfaces.* Journal of the American Statistical Association, vol. 89, no. 425, pages 53–64, 1994. 15

[LeCun 1998] Yann LeCun, Léon Bottou, Genevieve B Orr and Klaus-Robert Müller. *Efficient backprop.* In Neural networks: Tricks of the trade, pages 9–50. Springer, 1998. 8

[Lee 1998] Te-Won Lee. *Independent component analysis.* In Independent component analysis, pages 27–66. Springer, 1998. 15

[Lempitsky 2009] Victor S Lempitsky, Pushmeet Kohli, Carsten Rother and Toby Sharp. *Image segmentation with a bounding box prior.* In ICCV, pages 277–284. Citeseer, 2009. 11

[Li 2009] Li-Jia Li, Richard Socher and Li Fei-Fei. *Towards total scene understanding: Classification, annotation and segmentation in an automatic framework.* In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, pages 2036–2043. IEEE, 2009. 13

[Li 2012] C. Li, A. Kowdle, A. Saxena and T. Chen. *Toward Holistic Scene Understanding: Feedback Enabled Cascaded Classification Models.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 7, pages 1394–1408, July 2012. 1, 3, 4, 8, 10, 19, 22, 25, 29, 32, 33, 34, 35, 41, 42, 45, 50, 62, 63, 64

[Lin 2014] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár and C Lawrence Zitnick. *Microsoft coco: Common objects in context.* In European conference on computer vision, pages 740–755. Springer, 2014. 14

[Liu 2005] Luying Liu, Jianchu Kang, Jing Yu and Zhongliang Wang. *A comparative study on unsupervised feature selection methods for text clustering.* In Natural Language Processing and Knowledge Engineering, 2005. IEEE NLP-KE'05. Proceedings of 2005 IEEE International Conference on, pages 597–601. IEEE, 2005. 16

[Liu 2015] An-An Liu, Wei-Zhi Nie, Yu-Ting Su, Li Ma, Tong Hao and Zhao-Xuan Yang. *Coupled hidden conditional random fields for RGB-D human action recognition.* Signal Processing, vol. 112, pages 74–82, 2015. 9

[Maas 2013] Andrew L Maas, Awni Y Hannun and Andrew Y Ng. *Rectifier nonlinearities improve neural network acoustic models.* In Proc. icml, volume 30, page 3, 2013. 16

[Malfait 1997] Maurits Malfait and Dirk Roose. *Wavelet-based image denoising using a Markov random field a priori model.* IEEE transactions on image processing, vol. 6, no. 4, pages 549–565, 1997. 9

[Mwangi 2014] Benson Mwangi, Tian Siva Tian and Jair C Soares. *A review of feature reduction techniques in neuroimaging.* Neuroinformatics, vol. 12, no. 2, pages 229–244, 2014. 14

[Netrapalli 2014] Praneeth Netrapalli, UN Niranjan, Sujay Sanghavi, Animashree Anandkumar and Prateek Jain. *Non-convex robust PCA.* In Advances in Neural Information Processing Systems, pages 1107–1115, 2014. 15

[Oliva 2001] Aude Oliva and Antonio Torralba. *Modeling the shape of the scene: A holistic representation of the spatial envelope.* International journal of computer vision, vol. 42, no. 3, pages 145–175, 2001. 44, 45

[Paisitkriangkrai 2015] Sakrapee Paisitkriangkrai, Jamie Sherrah, Pranam Janney, Van-Den Hengel*et al.* *Effective semantic pixel labelling with convolutional networks and conditional random fields.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pages 36–43, 2015. 8, 9

[Park 2010] Dennis Park, Deva Ramanan and Charless Fowlkes. *Multiresolution models for object detection.* In European conference on computer vision, pages 241–254. Springer, 2010. 11

[Pearson 1901] Karl Pearson. *LIII. On lines and planes of closest fit to systems of points in space.* The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, vol. 2, no. 11, pages 559–572, 1901. 15

[Peng 2005] Hanchuan Peng, Fuhui Long and Chris Ding. *Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy.* IEEE Transactions on pattern analysis and machine intelligence, vol. 27, no. 8, pages 1226–1238, 2005. 15

[Phinyomark 2012] Angkoon Phinyomark, Pornchai Phukpattaranont and Chusak Limsakul. *Feature reduction and selection for EMG signal classification.* Expert Systems with Applications, vol. 39, no. 8, pages 7420–7431, 2012. 15

[Platt 1999] John Platt*et al.* *Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods.* Advances in large margin classifiers, vol. 10, no. 3, pages 61–74, 1999. 30

[Prest 2012] Alessandro Prest, Cordelia Schmid and Vittorio Ferrari. *Weakly supervised learning of interactions between humans and objects.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 3, pages 601–614, 2012. 11

[Quattoni 2005] Ariadna Quattoni, Michael Collins and Trevor Darrell. *Conditional random fields for object recognition.* In Advances in neural information processing systems, pages 1097–1104, 2005. 9

[Rabiner 1986] L. R. Rabiner and B. H. Juang. *An introduction to hidden Markov models.* IEEE ASSp Magazine, 1986. 9

[Rabinovich 2009] A. Rabinovich and S. Belongie. *Scenes vs. objects: A comparative study of two approaches to context based recognition.* In 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, pages 92–99, June 2009. 11

[Ross 2003] Arun Ross and Anil Jain. *Information fusion in biometrics.* Pattern recognition letters, vol. 24, no. 13, pages 2115–2125, 2003. 8

[Rowley 1998] Henry A Rowley, Shumeet Baluja and Takeo Kanade. *Neural network-based face detection.* IEEE Transactions on pattern analysis and machine intelligence, vol. 20, no. 1, pages 23–38, 1998. 7

[Russell 2008] Bryan C Russell, Antonio Torralba, Kevin P Murphy and William T Freeman. *LabelMe: a database and web-based tool for image annotation.* International journal of computer vision, vol. 77, no. 1-3, pages 157–173, 2008. 14

[Saeys 2007] Yvan Saeys, Iñaki Inza and Pedro Larrañaga. *A review of feature selection techniques in bioinformatics.* bioinformatics, vol. 23, no. 19, pages 2507–2517, 2007. 15

[Schölkopf 1997] Bernhard Schölkopf, Alexander Smola and Klaus-Robert Müller. *Kernel principal component analysis.* In International Conference on Artificial Neural Networks, pages 583–588. Springer, 1997. 15

[Sorzano 2014] Carlos Oscar Sánchez Sorzano, Javier Vargas and A Pascual Montano. *A survey of dimensionality reduction techniques.* arXiv preprint arXiv:1403.2877, 2014. 14, 15

[Souiai 2013] Mohamed Souiai, Claudia Nieuwenhuis, Evgeny Strekalovskiy and Daniel Cremers. *Convex optimization for scene understanding.* In Proceedings of the IEEE International Conference on Computer Vision Workshops, pages 9–14, 2013. 13

[Stowell 2014] Dan Stowell and Mark D Plumbley. *Automatic large-scale classification of bird sounds is strongly improved by unsupervised feature learning.* PeerJ, vol. 2, page e488, 2014. 16

[Sudderth 2005] Erik B Sudderth, Antonio Torralba, William T Freeman and Alan S Willsky. *Learning hierarchical models of scenes, objects, and parts.* In Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on, volume 2, pages 1331–1338. IEEE, 2005. 13

[Sudderth 2006] Erik B Sudderth, Antonio Torralba, William T Freeman and Alan S Willsky. *Depth from familiar objects: A hierarchical model for 3D scenes.* In null, pages 2410–2417. IEEE, 2006. 10

[Sun 2011] Min Sun and Silvio Savarese. *Articulated part-based model for joint object detection and pose estimation.* 2011. 12

[Sung 2009] Yun-Hsuan Sung and Dan Jurafsky. *Hidden conditional random fields for phone recognition.* In Automatic Speech Recognition & Understanding, 2009. ASRU 2009. IEEE Workshop on, pages 107–112. IEEE, 2009. 9

[Sutton 2005] Charles Sutton and Andrew McCallum. *Joint parsing and semantic role labeling.* In Proceedings of the Ninth Conference on Computational Natural Language Learning, pages 225–228. Association for Computational Linguistics, 2005. 12

[Thrun 2004] Sebastian Thrun, Christian Martin, Yufeng Liu, Dirk Hahnel, Rosemary Emery-Montemerlo, Deepayan Chakrabarti and Wolfram Burgard. *A real-time expectation-maximization algorithm for acquiring multiplanar maps of indoor environments with mobile robots.* IEEE Transactions on Robotics and Automation, vol. 20, no. 3, pages 433–443, 2004. 9, 10

[Torralba 2002] Antonio Torralba and Aude Oliva. *Depth estimation from image structure.* IEEE Transactions on pattern analysis and machine intelligence, vol. 24, no. 9, pages 1226–1238, 2002. 11

[Torralba 2003] Antonio Torralba. *Contextual priming for object detection.* International journal of computer vision, vol. 53, no. 2, pages 169–191, 2003. 11

[Torralba 2005] Antonio Torralba, Kevin P Murphy and William T Freeman. *Contextual models for object detection using boosted random fields.* In Advances in neural information processing systems, pages 1401–1408, 2005. 7

[Wang 2007] Yong Wang and Shaogang Gong. *Conditional Random Field for Natural Scene Categorization.* In BMVC, pages 1–10. Citeseer, 2007. 9

[Wang 2009] Yang Wang and Greg Mori. *Max-margin hidden conditional random fields for human action recognition.* In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, pages 872–879. IEEE, 2009. 9

[Wang 2013] Xiaoyang Wang and Qiang Ji. *A unified probabilistic approach modeling relationships between attributes and objects.* In Proceedings of the IEEE International Conference on Computer Vision, pages 2120–2127, 2013. 13

[Wang 2015] Shenlong Wang, Sanja Fidler and Raquel Urtasun. *Holistic 3d scene understanding from a single geo-tagged image.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3964–3972, 2015. 13

[Wei 2017] Ping Wei, Yibiao Zhao, Nanning Zheng and Song-Chun Zhu. *Modeling 4d human-object interactions for joint event segmentation, recognition, and object localization.* IEEE transactions on pattern analysis and machine intelligence, vol. 39, no. 6, pages 1165–1179, 2017. 2, 13

[Winn 2006] John Winn and Jamie Shotton. *The layout consistent random field for recognizing and segmenting partially occluded objects.* In Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, volume 1, pages 37–44. IEEE, 2006. 12

[Wojek 2008] Christian Wojek and Bernt Schiele. *A dynamic conditional random field model for joint labeling of object and scene classes.* In European Conference on Computer Vision, pages 733–747. Springer, 2008. 13

[Wu 2013] Xinxiao Wu, Dong Xu, Lixin Duan, Jiebo Luo and Yunde Jia. *Action recognition using multilevel features and latent structural SVM.* IEEE transactions on Circuits and Systems for Video Technology, vol. 23, no. 8, pages 1422–1431, 2013. 9

[Yang 1997] Yiming Yang and Jan O Pedersen. *A comparative study on feature selection in text categorization.* In Icml, volume 97, pages 412–420, 1997. 14

[Yao 2012] J. Yao, S. Fidler and R. Urtasun. *Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation.* In 2012 IEEE Conference on Computer Vision and Pattern Recognition, pages 702–709, June 2012. 2, 13

[Yi 2014] Won-Jae Yi, Oishee Sarkar, Sivisa Mathavan and Jafar Saniie. *Wearable sensor data fusion for remote health assessment and fall detection.* In Electro/Information Technology (EIT), 2014 IEEE International Conference on, pages 303–307. IEEE, 2014. 8

[Yu 2010] Dong Yu and Li Deng. *Deep-structured hidden conditional random fields for phonetic recognition.* In Eleventh Annual Conference of the International Speech Communication Association, 2010. 8, 9

[Zhang 2004] Zhenyue Zhang and Hongyuan Zha. *Principal manifolds and nonlinear dimensionality reduction via tangent space alignment.* SIAM journal on scientific computing, vol. 26, no. 1, pages 313–338, 2004. 15

[Zhang 2006] Cha Zhang, John C Platt and Paul A Viola. *Multiple instance boosting for object detection.* In Advances in neural information processing systems, pages 1417–1424, 2006. 7

[Zhang 2010] Jianguo Zhang and Shaogang Gong. *Action categorization with modified hidden conditional random field.* Pattern Recognition, vol. 43, no. 1, pages 197–203, 2010. 9

[Zhang 2014] Xueliang Zhang, Pengfeng Xiao, Xuezhi Feng, Jiangeng Wang and Zuo Wang. *Hybrid region merging method for segmentation of high-resolution remote sensing images.* ISPRS Journal of Photogrammetry and Remote Sensing, vol. 98, pages 19–28, 2014. 14

[Zhao 2013] Yibiao Zhao and Song-Chun Zhu. *Scene parsing by integrating function, geometry and appearance models.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3119–3126, 2013. 11

[Zhu 2010] Long Zhu, Yuanhao Chen, Alan Yuille and William Freeman. *Latent hierarchical structural learning for object detection.* In Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, pages 1062–1069. IEEE, 2010. 9

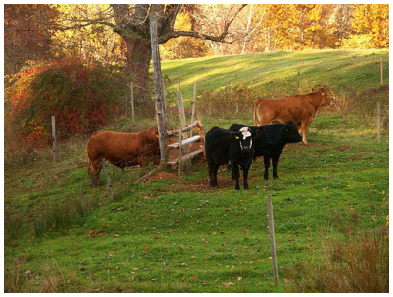
[Zweig 2007] Alon Zweig and Daphna Weinshall. Exploiting object hierarchy: Combining models from different category levels. IEEE, 2007. 11

Appendix A

Supplementary Material

A.1 Exemplary Images

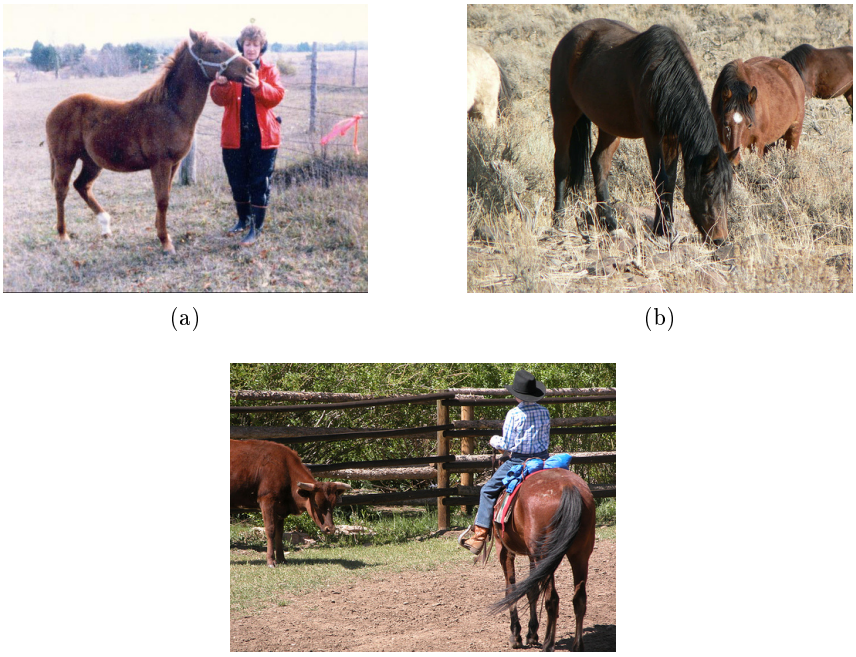
Figure A.1: Exemplary image of the disappearance of object instances during the resizing step. As the image is resized in such a way, that the included red box constitutes a single pixel in the resulting detection map, any object smaller than this box disappears. Consequently, the second cassification layer does not incorporate information of these instances during classification and is not able to provide reliable information about these instances during the feedback step.



(a)

(b)

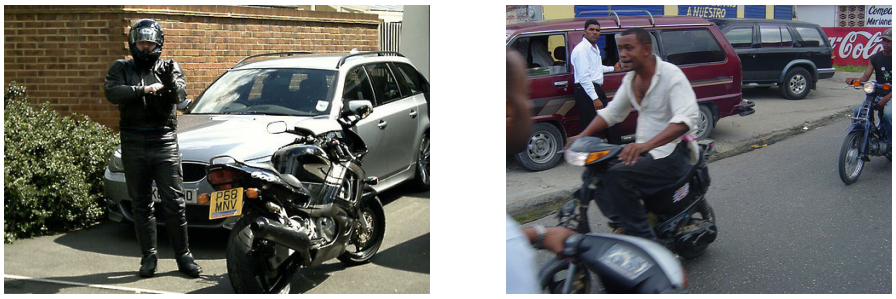
Figure A.2: Exemplary images for usual relationships of the class cow



(a) (b)

(c)

Figure A.3: Exemplary images for usual relationships of the class horse



(a) (b)

Figure A.4: Exemplary images for usual person-car relationships.

A.2 Plots - Explained Variance as a function of the number of principal components

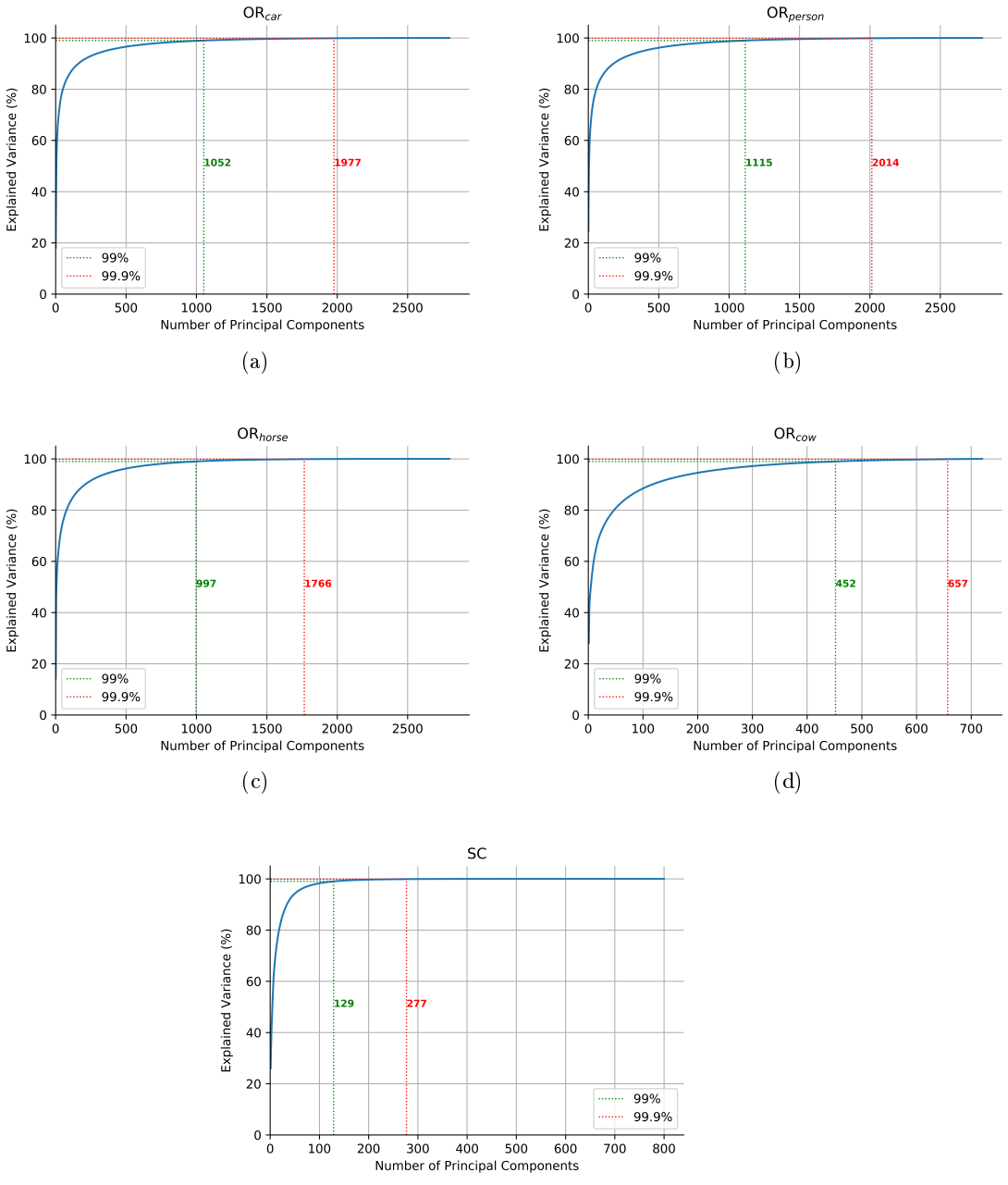


Figure A.5: Explained Variance as a function of the number of principal components for the Object Recognition (a)-(d) and the Scene Categorization Task (e).