Exploring Multi-Modal and Structured Representation Learning for Visual Image and Video Understanding

Dan Xu

A dissertation submitted in partial fulfilment
of the requirements for the degree of
Doctor of Philosophy
of the
University of Trento.

Advisor: Prof. Dr. Nicu Sebe,
ICT International Doctoral School,
Department of Information Engineering and Computer Science

April 2018

Contents

publication

This thesis consists of the following publications:

- Chapter 2:
  - **Dan Xu**, Xavier Alameda-Pineda, Jingkuan Song, Elisa Ricci, Nicu Sebe, "Academic Coupled Dictionary Learning for Sketch Based Image Retrieval", ACM International Conference on Multimedia (**ACM MM**), Amsterdam, Netherland, 2016 (**Oral, long paper**)
  - **Dan Xu**, Jingkuan Song, Xavier Alameda-Pineda, Elisa Ricci, Nicu Sebe, "Multi-Paced Dictionary Learning for Cross-Domain Retrieval and Recognition", 23rd International Conference on Pattern Recognition (**ICPR**), Cancun, Mexico, 2016 (**Oral, Best Scientific Paper**)

- Chapter 3:
  - **Dan Xu**, Wanli Ouyang, Elisa Ricci, Xiaogang Wang, Nicu Sebe, "Learning Cross-Modal Deep Representations for Robust Pedestrian Detection", IEEE Conference on Computer Vision and Pattern Recognition (**CVPR**), Hawaii, USA, 2017

- Chapter 4:
  - **Dan Xu**, Elisa Ricci, Wanli Ouyang, Xiaogang Wang, Nicu Sebe, "Multi-Scale Continuous CRFs as Sequential Deep Networks for Monocular Depth Estimation", IEEE Conference on Computer Vision and Pattern Recognition (**CVPR**), Hawaii, USA, 2017 (**Spotlight**).
  - **Dan Xu**, Elisa Ricci, Wanli Ouyang, Xiaogang Wang, Nicu Sebe, "Monocular Depth Estimation using Multi-Scale Continuous CRFs as Sequential Deep Networks", IEEE Transactions on Pattern Analysis and Machine Intelligence (**T-PAMI**) (in press), 2018

- Chapter 5:
  - **Dan Xu**, Wanli Ouyang, Xiaogang Wang, Nicu Sebe, "PAD-Net: Multi-Tasks Guided Prediction-and-Distillation Network for Simultaneous Depth Estimation and Scene Parsing", IEEE Conference on Computer Vision and Pattern Recognition (**CVPR**), Salt Lake City, USA, 2018.

The following papers are published during the course of the Ph.D but not included in this thesis:

- **Dan Xu**, Wei Wang, Hao Tang, Nicu Sebe, Elisa Ricci, "Structured Attention Guided Convolutional Neural Fields for Monocular Depth Estimation", IEEE Conference on Computer Vision and Pattern Recognition (**CVPR**), Salt Lake City, USA, 2018 (**Spotlight**).

- Dapeng Chen, **Dan Xu**, Hongsheng Li, Nicu Sebe, Xiaogang Wang, "Group Consistent Similarity Learning via Deep CRFs for Person Re-Identification", IEEE Conference on Computer Vision and Pattern Recognition (**CVPR**), Salt Lake City, USA, 2018 (**Oral**).

- Wei Wang, Xavier Alameda-Pineda, **Dan Xu**, Elisa Ricci, Nicu Sebe, "Every Smile is Unique: Landmark-Guided Diverse Smile Generation", IEEE Conference on Computer Vision and Pattern Recognition (**CVPR**), Salt Lake City, USA, 2018.

- **Dan Xu**, Wanli Ouyang, Xavier Alameda-Pineda, Elisa Ricci, Xiaogang Wang, Nicu Sebe, "Learning Deep Structured Multi-Scale Features using Attention-Gated CRFs for Contour Prediction", The Thirty-first Annual Conference on Neural Information Processing Systems (**NIPS**), Long Beach, USA, 2017.

- Xavier Alameda-Pineda, Andrea Pilzer, **Dan Xu**, Elisa Ricci, Nicu Sebe, "Viraliency: Pooling Local Virality", IEEE Conference on Computer Vision and Pattern Recognition (**CVPR**), Hawaii, USA, 2017

- Nannan Li, **Dan Xu**, Zhenqiang Ying, Zhihao Li, Ge Li, "Search Action Proposals via Spatial Actionness Estimation and Temporal Path Inference and Tracking", Asian Conference on Computer Vision (**ACCV**), Taibei, Tanwan, 2016

- **Dan Xu**, Elisa Ricci, Yan Yan, Jingkuan Song and Nicu Sebe, "Learning Deep Representations of Appearance and Motion for Anomalous Event Detection", British Machine Vision Conference (**BMVC**), Swansea, UK, 2015 (**Oral, 7% acceptance rate**)

- Xiantong Zhen, Feng Zheng, Ling Shao, Xianbin Cao, **Dan Xu**, "Supervised Local Descriptor Learning for Human Action Recognition", IEEE Transactions on Multimedia(**T-MM**), 10.1109/TMM.2017.2700204

- **Dan Xu**, Yan Yan, Elisa Ricci, Nicu Sebe, "Detecting Anomalous Events in Videos by Learning Deep Representations of Appearance and Motion", Computer Vision and Image Understanding (**CVIU**),10.1016/j.cviu.2016.10.010.

CHAPTER 1

Introduction

Nowadays digital images and videos are playing an increasingly important role in our everyday lives. We are all producers and consumers of the image and video data, and people spend hundreds of millions of hours every day in watching and sharing the images and videos on various social medias such as YouTube, Facebook or Snapchat. All that is arguably just the beginning. In the next few years, the world would become more data-driven and more intensively connected through a proliferation of smart devices, ranging from mobiles, always-on home cameras to autonomous vehicles. The cameras on many of these devices will provide visual sensors to perceive, understand and navigate the world. As the explosive growth of the visual data, it is clear that human-powered visual understanding is limited and not scalable to deal with a large amount of data.

To tackle the challenge, it requires us to develop automatic or semi-automatic techniques which are capable of efficiently process, retrieve, detect, recognize and interact with the big visual data consisting of real-world images and videos. Many efforts have been made in recent years from both the academia and the industry to build highly effective and large-scale intelligent visual processing algorithms and systems for the target. One of the core aspects in the research line is how to learn robust representations from the data to better describe the data. In this thesis we study the problem of visual image and video understanding and specifically, we address the problem via designing and implementing novel multi-modal and structured representation learning approaches, both of which are fundamental research hot-spots in machine learning. Multi-modal representation learning [103] involves relating information from multiple input sources, and the structured representation learning [136] works on exploring rich structural information hidden in the data for robust feature learning. We investigate both the shallow representation learning frameworks such as dictionary learning and the deep representation learning frameworks such as deep neural networks. An illustration of the thesis is depicted in Figure 1.1, which shows the primary modules devised in our works, consisting of cross-paced representation learning, cross-modal feature learning and transferring, multi-scale structured prediction and fusion, multi-modal prediction and distillation. These approaches are further applied in different visual understanding topics, *i.e.* sketch-based-image retrieval (SBIR), video pedestrian detection, monocular depth estimation and scene parsing.

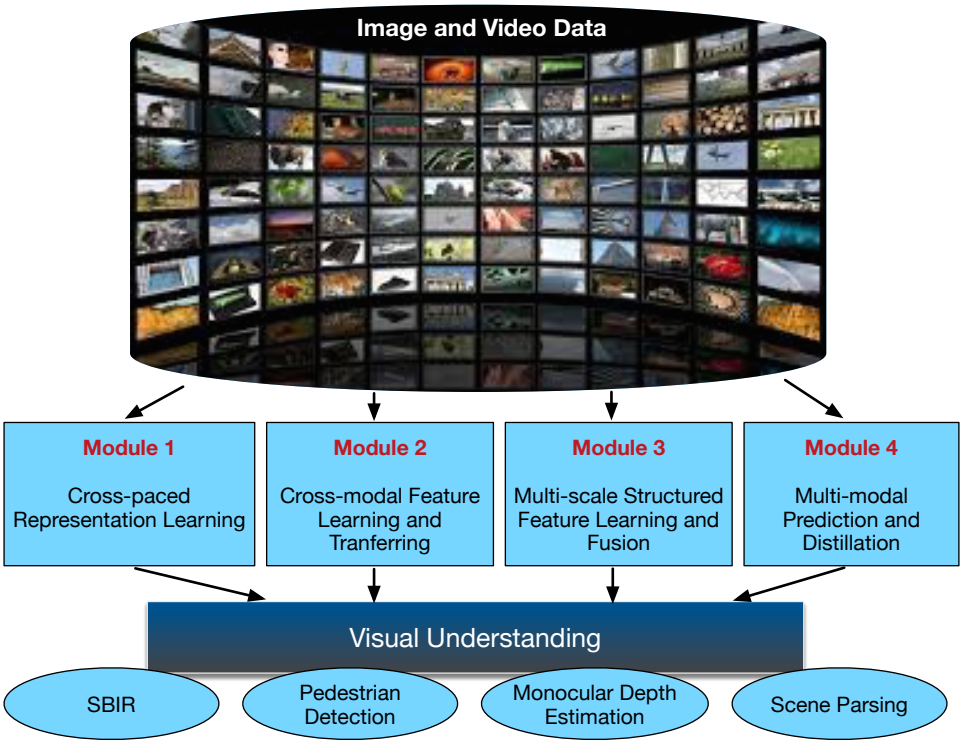


Figure 1.1: Overview of the proposed multi-modal and structured learning framework.

We start from introducing a cross-modal representation learning approach based on a coupled dictionary learning framework for the SBIR task. The most important motivation of the work is that, dictionary-based cross-domain representation learning methods are typically cast into non-convex minimization problems which are difficult to optimize, thus leading to unsatisfactory performance of the representations. Inspired by self-paced learning, a learning methodology designed to overcome convergence issues related to local optima by exploiting the samples in a meaningful order (*i.e.* easy to hard), we propose a novel cross-paced partial curriculum learning (CPPCL) framework. We present the work in Chapter 2, which was previously published by ACM Multimedia [160] and International Conference on Pattern Recognition [166]. We demonstrate the effectiveness of the proposed representation learning strategy and show superior performance on SBIR on four different publicly available datasets.

The Chapter 2 works on shallow representation learning. In Chapter 3, we further explore the cross-modal representation learning using deep neural networks, since deep learning has shown remarkable success in various computer vision tasks. Our main aim lies in two aspects: (i) we expect to learn the representations using an unsupervised setting as the annotation is costly for the big data; (ii) we want to transfer the learned cross-modal representations for a detection task while only data from one modality are required in the testing phase. To achieve the target, we propose a deep learning approach in Chapter 3, which was published by IEEE Conference on Computer Vision and Pattern Recognition [162] and is based on two main phases. First, given a multimodal dataset, a deep convolutional network is employed to reconstruct one modality from the other one. Then, the feature representations from the middle hidden layers of the reconstruction network are transferred to a second deep network for the detection task. While the proposed framework is generic for different vision problems, we apply the proposed approach for

pedestrian detection under bad illumination conditions. Specifically, the reconstruction network learns cross-modal representations from RGB and Thermal domains, and in the testing, only RGB data are used.

The multi-scale deep representations have been demonstrated very effective in many computer vision tasks. In Chapter 3, the reconstruction network learns a non-linear mapping from one modality to another to generate cross-modal representations. However, the multi-scale representations are not considered to be effectively fused for boosting the performance. Could we perform structured learning and fusion on the multi-scale features to obtain more effective representations? We work on this question in Chapter 4 via proposing a multi-scale CRF model and implementing it as a sequential neural network for end-to-end optimization, which is an extension of our published papers at IEEE Conference on Computer Vision and Pattern Recognition [164, 167]. The proposed approach is useful for continuous regression problems, and is applied in a monocular depth estimation task in our work. We effectively reconstruct the depth domain from the RGB domain via fusing the learned multi-scale structured deep predictions, and achieve state-of-the-art results on several publicly available datasets.

Finally, we study learning deep multi-modal representations from a single modality and utilizing them to facilitate multi-task predictions in a joint deep network in Chapter 5. The work was published by IEEE Conference on Computer Vision and Pattern Recognition [163]. The common difficulty of deep multi-task prediction is that the network is hard to have generalization ability on all tasks, which usually leads to worse performance on some of the tasks. To tackle this problem, our first novelty is to produce rich multi-modal data from intermediate tasks using one single modality as input and then utilize them for the target tasks. This new paradigm is not explored in existing works. Our second novelty is closely related to the first. There are two difficulties in the new paradigm. One is how to effectively use the multi-modal data obtained from intermediate auxiliary predictions. Our second novelty target at this point, while Cross-stitch Net [99], Sluice Net [120], and deep relation net [95], assume only single-modal data and thus do not consider it. The other difficulty is, after the features from multi-modal data are extracted, how to design a good network architecture so that the network communicates or shares features for different tasks. We perform simultaneous depth estimation and scene parsing tasks using the proposed approach, and consistently show significant performance gain on both tasks.

CHAPTER 2

Cross-Paced Representation Learning with Partial Curricula for Sketch-based Image Retrieval [1]

In this chapter we address the problem of learning robust cross-domain representations for sketch-based image retrieval (SBIR). While most SBIR approaches focus on extracting low- and mid-level descriptors for direct feature matching, recent works have shown the benefit of learning coupled feature representations to describe data from two related sources. However, cross-domain representation learning methods are typically cast into non-convex minimization problems that are difficult to optimize, leading to unsatisfactory performance. Inspired by self-paced learning, a learning methodology designed to overcome convergence issues related to local optima by exploiting the samples in a meaningful order (*i.e.* easy to hard), we introduce the cross-paced partial curriculum learning (CPPCL) framework. Compared with existing self-paced learning methods which only consider a single modality and cannot deal with prior knowledge, CPPCL is specifically designed to assess the learning pace by jointly handling data from dual sources and modality-specific prior information provided in the form of partial curricula. Additionally, thanks to the learned dictionaries, we demonstrate that the proposed CPPCL embeds robust coupled representations for SBIR. Our approach is extensively evaluated on four publicly available datasets (*i.e.* CUFS, Flickr15K, QueenMary SBIR and TU-Berlin Extension datasets), showing superior performance over competing SBIR methods.

2.1 Introduction

In the last few years, the developments in mobile device applications have increased the demand for powerful and efficient tools to query large-scale image databases. In particular, favored by the widespread diffusion of consumer touchscreen devices, sketch-based image retrieval (SBIR) has gained popularity. Most prior works on SBIR [55, 121, 33, 122, 123] focused on designing low- and mid-level features, and used the same type of descriptors for

[1] Dan Xu, Xavier Alameda-Pineda, Jingkuan Song, Elisa Ricci, Nicu Sebe, "Academic Coupled Dictionary Learning for Sketch Based Image Retrieval", ACM International Conference on Multimedia (ACM MM 2016). Dan Xu, Jingkuan Song, Xavier Alameda-Pineda, Elisa Ricci, Nicu Sebe, "Multi-Paced Dictionary Learning for Cross-Domain Retrieval and Recognition", 23rd International Conference on Pattern Recognition (ICPR 2016).

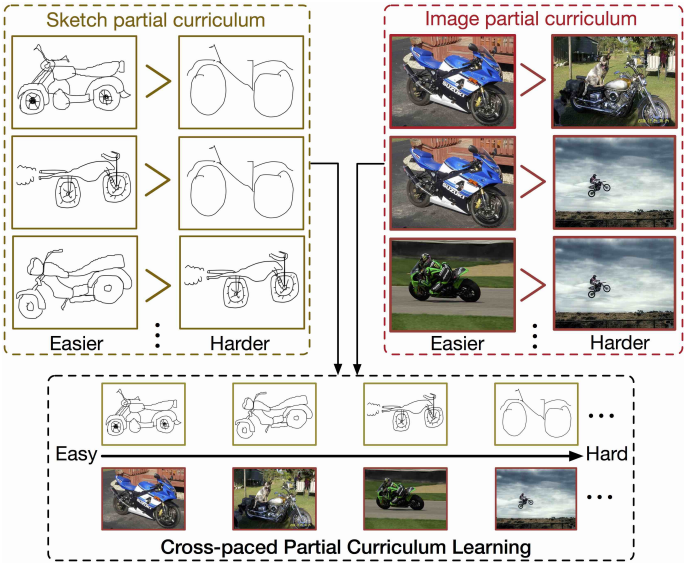


Figure 2.1: In real SBIR scenarios, both sketches and images show a wide range of visual complexity. Defining *a-priori* a full learning order (*i.e.* for all samples) based on the difficulty of the sketches/images is extremely challenging. Cross-paced partial curriculum learning combines the flexibility of partial modality-specific curricula with the power of self-paced learning strategies to automatically construct a full sample learning order that evolves over time until all training samples are used for learning.

representing both sketches and image edge maps, allowing a direct matching between the two modalities. However, these methods implicitly assume that the statistical distributions of image edges and sketches are similar. Unfortunately, this assumption does not hold in many applications. Therefore, more recent studies proposed to use different feature descriptors to better represent the different modalities and learned a shared feature space using cross-domain representation learning methods. In particular, recent approaches based on dictionary learning (DL) [79, 174, 154, 56] or deep networks [36, 149] have been proven especially successful for learning coupled representations from cross-modal data. However, these methods are usually based on non-convex optimization problems and can get easily stuck into local optima, with an adverse impact on the representational power and generalization capabilities of the learned descriptors.

Recent research efforts to overcome the problems associated to local optima resulted in two orthogonal trends: self-paced learning (SPL) [73] and curriculum learning (CL) [9]. The common denominator of both SPL and CL is to build a learning model with the help of a sample order reflecting the inherent data complexity. The rationale is that, when this order is appropriately chosen, we increase the chances of avoiding local minima. SPL and CL have been successfully applied to several computer vision tasks, such as object tracking [139] and visual category discovery [80]. Even if both strategies share a common denominator, they are quite different in spirit. Indeed, while in CL the learning order is pre-determined by an expert or according to other prior knowledge (*e.g.* extracted from the data), in SPL the algorithm automatically assesses the learning order usually based on the feedback of the learned model. Recently, Jiang *et al.* [62] demonstrated that further advantages in terms of performance can be obtained by combining CL and SPL.

The particular case of SBIR is of special interest regarding CL, SPL and possible combinations. Indeed, as shown in Fig. 2.1, the visual complexity of sketches and images

greatly varies, and methods attempting to exploit these variations would a priori have more chances to successfully learn efficient and robust cross-domain representations. Specifically, natural images are characterized by cluttered background and objects-of-interest captured at different scales or various poses. Similarly, sketches drawn by expert/non-expert show remarkable variations. Therefore, our aim is to turn what could be seen as an adversity, into an exploitable feature inherent to the data. However, there are two major problems which hinder the direct application of existing SPL and CL methods into cross-domain representation learning models for SBIR. Firstly, the SBIR task involves data from two different modalities, while most of the previous SPL and CL approaches are fundamentally designed to model data from a single modality. Secondly, CL methods assume the existence of a full curriculum (*i.e.* a complete order of all samples). This limits the applicability of CL methods to small/medium-scale problems, since the curriculum is usually designed by humans and assessing the easiness order of all samples (images and sketches) would be a chimerically resource-consuming task.

To address these problems, we design a novel cross-modality representation learning paradigm and apply it to the SBIR task. In details, we propose a novel self-paced learning strategy able to handle cross-modal data and to incorporate incomplete prior knowledge (*i.e.* partial modality-specific curricula), and we name it Cross-Paced Partial Curriculum Learning (CPPCL). Furthermore, we embed this strategy into a coupled dictionary learning framework for computing robust cross-domain representations. Specifically, our method learns a pair of image- and sketch-specific dictionaries, together with the associated sparse codes, enforcing the similarity between the codes of corresponding sketches and images. The reconstruction loss with the learned dictionaries, the code correspondence and the partial modality-specific curricula jointly determine which samples to learn from. We extensively evaluate our cross-domain representation learning on four publicly available datasets (*i.e.* CUFS, Flickr15K, QueenMary SBIR, TU-Berlin Extension), demonstrating the effectiveness of the proposed learning strategy and achieves superior performance over competing SBIR approaches. The main contributions of this paper are:

- We introduce the cross-paced partial curriculum learning paradigm to effectively integrate the self-pacing philosophy with modality-specific partial curricula and investigate different self-paced regularizers.

- We propose an instantiation of CPPCL within the framework of coupled dictionary learning to obtain robust cross-domain representations for SBIR and we develop an efficient algorithm to learn the modality-specific dictionaries and codes, while assessing the optimal learning order jointly from the partial curricula and the representation power of the model at the current iteration.

- We carry out an extensive experimental evaluation and analysis of the whole cross-domain representation learning framework, exhibiting its effectiveness for SBIR on four different publicly available datasets.

The paper extends our conference submission [160] by reformulating the proposed CPPCL considering different self-paced regularization terms (*e.g.* adding Self-paced regularizer A in Section 2.3.2) and developing the associated optimization algorithms (Section 2.4). From the experiments perspective, we discuss the influence, similarities and differences when using the different regularizing schemes within the proposed cross-paced learning

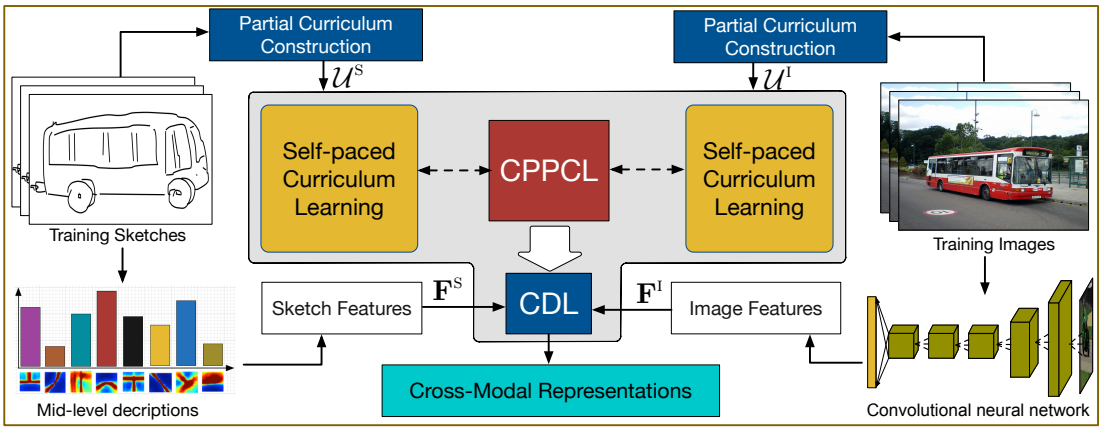


Figure 2.2: Overview of the proposed cross-modal representation learning method. Features extracted from sketches (*e.g.* LKS descriptors) and images (*e.g.* CNN-derived representations) are employed within a coupled dictionary learning (CDL) framework for computing cross-modal representations for SBIR. Our CDL integrates a novel cross-paced partial curriculum learning paradigm which allows the learning algorithm to start with easy samples and gradually involve hard samples according to predetermined heuristics (*i.e.* modality-specific partial curricula).

framework on two publicly available datasets. A more in-depth analysis is conducted to further show the effectiveness of the proposed approach, including some parameter sensitivity study and a convergence analysis of different models (Section 2.5). Moreover, the introduction and related works parts are reorganized and significantly extended.

The rest of the paper is organized as follows: we first review the related work in Section 2.2, and then elaborate the details of the proposed approach and associated optimization algorithms in Sections 2.3 and 2.4 respectively. The experimental results are presented in Section 4.5 and we conclude the paper in Section 4.6.

2.2 Related Work

This section reviews related works in the areas of: (i) sketch-based image retrieval, (ii) self-paced and curriculum learning and (iii) cross-domain dictionary learning.

2.2.1 Sketch-based Image Retrieval

SBIR approaches are mostly based on matching feature descriptors of the query sketch with those of the edge maps associated to the images in the database. Early works on SBIR attempted to use existing low-level feature representations (*e.g.* describing color, texture, contour and shape) for both the sketch and the image modalities. Both global low-level descriptors (*e.g.* color histograms [65], distribution of edge pixels [17], elastic contours [11]) and local ones (*e.g.* spark descriptors [34], SYM-FISH [16], SIFT [96], HOG [24]) were investigated in the literature. Other works focused on developing specific descriptors for SBIR. For instance, Hu *et al.* [55] introduced the Gradient Field HOG (GF-HOG) descriptor, extending HOG to better represent sketches, and constructed a large dataset for evaluation: the Flickr15K. Saavendra *et al.* [123] also proposed a modified version of HOG, the histogram of edge local orientations (HELO), to tackle the problem of sparsity arising when HOG descriptors are applied to sketches.

To represent sketches or image edges more robustly, most recent SBIR methods focused on constructing mid- or high-level feature descriptors. Several works considered the bag-of-words (BoW) technique to aggregate low-level features and generate mid-level representations [34, 138, 87]. In addition to BoW-based methods, other approaches also

focused on mid-level representations. For instance, in [122] an effective method to generate mid-level patterns, named learned keyshapes (LKS), was proposed for representing sketches. Yi *et al.*[84] built mid-level representations for both sketches and images by optimizing a deformable part-based model. Xiao *et al.*[157] designed a shape feature descriptor especially useful for preserving the shape information of sketches. A perceptual grouping framework was introduced in [114] to organize image edges into a meaningful structure and was adopted for generating human-like sketches useful for SBIR. Yu *et al.*[179, 178] proposed to adopt deep CNNs to learn high-level sketch representations. Similarly, Liu *et al.*[91] explored deep representations within a binary coding framework for fast sketch based image retrieval.

In all these works the same low- and mid/high-level representations are used to describe both the sketch and image modalities, such as to facilitate direct feature matching. However, due to the difference in appearance between sketches and images, different features are more suitable to represent the two modalities. Following this idea, some works proposed learning a shared feature space for the two modalities [151, 170]. However, none of these works considered exploiting the visual complexity of samples to learn more effective cross-modal feature representations.

2.2.2 Self-paced and Curriculum Learning

Inspired by the way the human brain explores the world, *i.e.* starting from easy concepts first and gradually involving more complex notions, self-paced learning [73] and curriculum learning [9] have been recently developed. The idea of SPL and CL is to learn models in an incremental fashion from samples with variate difficulty presented in a meaningful order. Due to their generality, these techniques have been considered in a broad spectrum of learning tasks and models, including matrix factorization [187, 62], clustering [159], multi-task learning [111] and dictionary learning [141, 170]. They have also shown to be successful in many computer vision applications such as object tracking [139], media retrieval [60], visual category discovery [80] and event detection [61].

Although self-paced learning and curriculum learning develop from the same rationale, they differ in the specific implementation schemes. In CL, the learning order (*i.e.* the curriculum) is pre-defined according to prior knowledge and fixed during the learning phase, while in SPL the curriculum is dynamically determined based on the feedback from the learner. Since the sample order in SPL is dynamically inferred, one challenging task is to design a meaningful strategy of assessing the difficulty of the training samples. Previous works have addressed this issue in different ways. The most common strategy is to measure the easiness of a sample by computing the associated loss [73]. Alternatively, Jiang *et al.*[60] proposed to take into account the dissimilarity with respect to what has already been learned. To incorporate the benefits of both SPL and CL, a recent work [62] proposed a self-paced curriculum learning framework in which the learning order is jointly determined by a predefined full-order curriculum and the learning feedback. However, none of these previous works focused on handling multi-modal data. Our approach not only extends the self-paced learning paradigm to cope with cross-domain data, but, more importantly, is naturally able to utilize domain specific partial ordering information. In fact, opposite to the method in [62] which needs a full-order curriculum, our approach integrates prior knowledge in a form of partial curriculum. Thus, it can be applied to large scale (SBIR) tasks.

2.2.3 Cross-domain Dictionary Learning

Dictionary learning [79] is a popular method for finding effective sparse representations of input data. DL has been successfully applied in various image processing and computer vision tasks, such as image denoising [98] and video event detection [172]. With the fast emergence of large scale cross-domain datasets, traditional DL approaches have been extended to cross-modal tasks. For instance, Yang *et al.*[174] proposed to learn a set of source-specific dictionaries from samples corresponding to different domains in a coupled fashion in the context of image super-resolution. In [154] Wang *et al.*introduced semi-coupled DL for photo-sketch synthesis, where source-specific dictionaries are learned together with a mapping function which describes the intrinsic relationship between domains. Similarly, Huang and Wang [56] proposed a framework to simultaneously learn a pair of domain-specific dictionaries and the associated representations. Coupled DL approaches have also been applied to SBIR both in [154] and [56] and to other related tasks, such as sketch-based 3D object retrieval [150] and sketch recognition [43]. However, none of these cross-domain DL methods explore self-paced learning or curriculum learning to construct more robust features.

2.3 The Proposed Approach

As discussed in Section 3.1, in this chapter we introduce a novel cross-domain representation learning framework for sketch-based image retrieval. Figure 5.2 shows an overview of our approach. The overall objective of the proposed model is to learn robust cross-modal feature representations. As previously mentioned, commonly used cross-modal representation leaning methods, such as coupled dictionary learning [174] and multi-modal deep learning [103], usually rely on non-convex optimization problems and are likely to get stuck at a bad local optimal. We investigate how to incorporate the ideas of SPL and partial curriculum learning within a principled unified dictionary-based learning framework.

In the following, we describe the proposed approach in details, presenting the general formulation of the overall learning problem (Section 2.3.1), the details of CPPCL (Section 2.3.2), the instantiation of CPPCL into CDL (Section 2.3.3) and the construction of modality-specific curricula (Section 2.3.4).

2.3.1 Problem Formulation

Let us assume the existence of K sketches and denote the features extracted from the k-th sketch as $\mathbf{f}_k^{\mathrm{S}} \in \mathbb{R}^{m_{\mathrm{S}}}$. Similarly, we assume the existence of L images and denote the features extracted from the l-th image as $\mathbf{f}_l^{\mathrm{I}} \in \mathbb{R}^{m_{\mathrm{I}}}$. Each sketch (resp. image) corresponds to a new cross-modal representation to be learned, denoted as $\mathbf{c}_k^{\mathrm{S}} \in \mathbb{R}^N$ (resp. $\mathbf{c}_l^{\mathrm{I}} \in \mathbb{R}^N$) with N being the dimension of the new representation. We also define $\mathbf{F}^{\mathrm{S}} = [\mathbf{f}_1^{\mathrm{S}}, \ldots, \mathbf{f}_K^{\mathrm{S}}] \in \mathbb{R}^{m_{\mathrm{S}} \times K}$ as the matrix of all sketch features, and $\mathbf{F}^{\mathrm{I}}$, $\mathbf{C}^{\mathrm{S}}$ and $\mathbf{C}^{\mathrm{I}}$ analogously. We denote $\mathcal{U}^{\mathrm{S}}$ and $\mathcal{U}^{\mathrm{I}}$ as the modality-specific partial curricula constructed from the sketch and the image domains respectively. The overall learning objective of the proposed cross-paced representation learning with partial curricula model can be written as:

$$\begin{aligned} \min_{\mathbf{C}^{\mathrm{S}},\mathbf{C}^{\mathrm{I}},\mathbf{V}^{\mathrm{J}},\boldsymbol{\xi}^{\mathrm{J}}} \; & \mathcal{L}_{\mathrm{RL}}(\mathbf{C}^{\mathrm{S}},\mathbf{C}^{\mathrm{I}},\mathbf{V}^{\mathrm{J}};\mathbf{F}^{\mathrm{S}},\mathbf{F}^{\mathrm{I}}) \\ & + f_{\mathrm{PC}}(\boldsymbol{\xi}^{\mathrm{J}};\mathcal{U}^{\mathrm{S}},\mathcal{U}^{\mathrm{I}}) + f_{\mathrm{SP}}(\mathbf{V}^{\mathrm{J}};\gamma) \\ \text{s.t.} \quad & v_k^{\mathrm{S}}, v_l^{\mathrm{I}} \in \{0,1\} \quad \forall k,l \end{aligned} \tag{2.1}$$

where $\mathbf{V}^{\mathrm{J}} = \mathrm{diag}(\mathbf{V}^{\mathrm{S}}, \mathbf{V}^{\mathrm{I}})$ with $\mathbf{V}^{\mathrm{S}} = \mathrm{diag}(v_1^{\mathrm{S}}, ..., v_K^{\mathrm{S}})$ and $\mathbf{V}^{\mathrm{I}} = \mathrm{diag}(v_1^{\mathrm{I}}, ..., v_L^{\mathrm{I}})$, are binary pacing variables which indicate whether a training instance (sketch or image) has to be used for learning or not. $\mathcal{L}_{\mathrm{RL}}(\mathbf{C}^{\mathrm{S}}, \mathbf{C}^{\mathrm{I}}, \mathbf{V}^{\mathrm{J}}; \mathbf{F}^{\mathrm{S}}, \mathbf{F}^{\mathrm{I}})$ is a cross-modal representation learning term given $\mathbf{F}^{\mathrm{S}}$ and $\mathbf{F}^{\mathrm{I}}$. For the proposed learning framework, this term is flexible to employ various representation learning methods such as coupled dictionary learning [56], cross-domain subspace learning [151] and deep learning [36]. $f_{\mathrm{SP}}(\mathbf{V}^{\mathrm{J}}; \gamma)$ is a cross-modal self-paced regularizer determining the learning order of samples in two modalities, and $\gamma \geq 0$ is a self-paced parameter which controls the learning pace. $f_{\mathrm{PC}}(\boldsymbol{\xi}^{\mathrm{J}}; \mathcal{U}^{\mathrm{S}}, \mathcal{U}^{\mathrm{I}})$ is a partial curriculum (PC) regularizer which makes the learning order match with the pre-determined modality-specific curricula $\mathcal{U}^{\mathrm{S}}$ and $\mathcal{U}^{\mathrm{I}}$ as much as possible, and $\boldsymbol{\xi}^{\mathrm{J}}$ represent partial curriculum learning variables. In the following, we present the details of the proposed learning framework.

2.3.2 Cross-paced Partial Curriculum Learning

CPPCL is a joint learning paradigm which combines a self-paced and a partial curriculum learning scheme, corresponding to the two components $f_{\mathrm{PC}}(\boldsymbol{\xi}^{\mathrm{J}}; \mathcal{U}^{\mathrm{S}}, \mathcal{U}^{\mathrm{I}})$ and $f_{\mathrm{SP}}(\mathbf{V}^{\mathrm{J}}; \gamma)$ as described in Eqn. 2.1. By doing so, the learning order is simultaneously determined by the pre-defined prior knowledge (*i.e.* partial-order modality-specific curriculum) and the feedback from the learner during training.

As mentioned in Section 2.3.1, in the self-paced learning philosophy, there is a pacing binary variable $v_k^{\mathrm{S}} \in \{0, 1\}$ (respectively $v_l^{\mathrm{I}} \in \{0, 1\}$) associated to sketch k (respectively to image l), determining the learning order of the training samples. Importantly, v_k^{S} and v_l^{I} are not fixed and evolve during the training phase. To analyze the influence of the self-paced learning scheme, we investigate two different self-paced regularizers in our learning framework.

Self-paced regularizer A

is proposed to take into account the diversity of training data. We assume that the training data of the sketch modality are split into G^{S} groups or classes (either learned from the data or provided in advance). We define a group-specific indicator vector $\mathbf{p}_i^{\mathrm{S}} \in \mathbb{R}^K$, where $p_{i,k}^{\mathrm{S}} = 1$ if and only if sample k belongs to group i ($i \in \{1, ..., G^{\mathrm{S}}\}$), and $p_{i,k} = 0$ otherwise. We devise a penalty over $\mathbf{V}^{\mathrm{S}}$ that is normalized over the groups' size, denoted by E_i^{S}. The definitions in the image domain, *i.e.* for G^{I}, $\mathbf{p}_j^{\mathrm{I}}$ and E_j^{I} are analogous. The regularizer writes:

$$f_{\mathrm{SP}_A}(\mathbf{V}^{\mathrm{J}}; \gamma) = -\gamma \left(\sum_{i=1}^{G^{\mathrm{S}}} \frac{1}{E_i^{\mathrm{S}}} \|\mathbf{V}^{\mathrm{S}} \mathbf{p}_i^{\mathrm{S}}\|_1 + \sum_{j=1}^{G^{\mathrm{I}}} \frac{1}{E_j^{\mathrm{I}}} \|\mathbf{V}^{\mathrm{I}} \mathbf{p}_j^{\mathrm{I}}\|_1 \right). \tag{2.2}$$

This term enforces learning from different groups/classes and therefore it is closely related to SPL with diversity [61]. Similarly to [61], the idea is to learn not only from easy samples as in the standard SPL [73] but also from samples that are dissimilar from what has already been learned. However, with respect to [61], the proposed regularizer has two prominent advantages: (i) we avoid using group norms that significantly increase the complexity of the optimization solvers and (ii) we introduce the normalization factors E_i^{S} and E_j^{I} that soften the bias induced by dissimilar group cardinalities.

Self-paced regularizer B

introduces a slight modeling change. Indeed, following Zhao *et al.* [187] we consider the self-pacing variables v_k^{S} and v_l^{I} to be continuous in the range $[0, 1]$. With this choice, we allow the model to take a soft decision and assess the importance of the training sample,

rather than force the method to choose between using/ignoring the sample at the current iteration. Notice that the previous self-paced regularizer (f_{SP_A}) can also be used with continuous self-pacing variables. In addition, considering v_k^{S} and v_l^{I} to be continuous opens the door to the definition of more sophisticated self-pacing regularizers such as:

$$f_{\mathrm{SP}_B}(\mathbf{V}^{\mathrm{J}};\gamma) = -\frac{\gamma}{2}\left(\sum_{k=1}^{K} Q(v_k^{\mathrm{S}}) + \sum_{l=1}^{L} Q(v_l^{\mathrm{I}})\right), \tag{2.3}$$

where $Q(v) = v^2 - 2v$ as in [187].

Importantly, the penalty induced by the regularizer evolves over time so as to incorporate more and more samples to be part of the training set. Specifically, the self-paced parameter γ is multiplied by a step size η ($\eta > 1$) in order to increase γ at each iteration, as in traditional SPL methods [73]. This is done for both f_{SP_A} and f_{SP_B} regularizers.

An important methodological contribution of our work is to include **modality-specific partial curricula** into a representation learning framework and to study its behavior within the SPL strategy already discussed. Subsequently, we assume the existence of two modality-specific sets of constraints $\mathcal{U}^{\mathrm{S}}$ and $\mathcal{U}^{\mathrm{I}}$. Each element of the sets consists of an index pair representing that if $(k,k') \in \mathcal{U}^{\mathrm{S}}$, then $v_k^{\mathrm{S}} < v_{k'}^{\mathrm{S}}$ and learning should be performed considering a priori $\mathbf{f}_{k'}^{\mathrm{S}}$ before $\mathbf{f}_k^{\mathrm{S}}$, as it corresponds to an easier sample. Depending on the way the curricula are constructed $\mathcal{U}^{\mathrm{S}}$ could contain incompatibilities, for instance, $\{(k,k'),(k',k''),(k'',k)\} \subset \mathcal{U}^{\mathrm{S}}$. In addition, the cross-modal terms could also induce incompatibilities between the two modalities. Therefore, it is desirable to relax the constraints using a set of slack variables $\xi_{kk'}^{\mathrm{S}}$, $\xi_{ll'}^{\mathrm{I}}$, and the partial curriculum regularizer is written as:

$$f_{\mathrm{PC}}(\boldsymbol{\xi}^{\mathrm{J}};\mathcal{U}^{\mathrm{S}},\mathcal{U}^{\mathrm{I}}) = \mu\left(\sum_{(k,k')\in\mathcal{U}^{\mathrm{S}}} \xi_{kk'}^{\mathrm{S}} + \sum_{(l,l')\in\mathcal{U}^{\mathrm{I}}} \xi_{ll'}^{\mathrm{I}}\right), \tag{2.4}$$

where $\boldsymbol{\xi}^{\mathrm{J}} = [[\xi_{kk'}^{\mathrm{S}}]_{(k,k')\in\mathcal{U}^{\mathrm{S}}}[\xi_{ll'}^{\mathrm{I}}]_{(l,l')\in\mathcal{U}^{\mathrm{I}}}]$ is the vector of all slack variables and f_{PC} is the partial curricula regularizer regulated by the parameter $\mu \geq 0$. In all, the optimization problem of CPPCL writes:

$$\begin{aligned}
\min_{\mathbf{V}^{\mathrm{J}},\boldsymbol{\xi}^{\mathrm{J}}} \; & f_{\mathrm{PC}}(\boldsymbol{\xi}^{\mathrm{J}};\mathcal{U}^{\mathrm{S}},\mathcal{U}^{\mathrm{I}}) + f_{\mathrm{SP}}(\mathbf{V}^{\mathrm{J}};\gamma) \\
& v_k^{\mathrm{S}}, v_l^{\mathrm{I}} \in \{0,1\} \quad \forall k,l, \\
& v_k^{\mathrm{S}} - v_{k'}^{\mathrm{S}} < \xi_{kk'}^{\mathrm{S}}, \xi_{kk'}^{\mathrm{S}} \geq 0, \forall (k,k') \in \mathcal{U}^{\mathrm{S}} \\
& v_l^{\mathrm{I}} - v_{l'}^{\mathrm{I}} < \xi_{ll'}^{\mathrm{I}}, \xi_{ll'}^{\mathrm{I}} \geq 0, \forall (l,l') \in \mathcal{U}^{\mathrm{I}}.
\end{aligned}$$

2.3.3 Instantiation of CPPCL into CDL

To learn cross-modal representations for SBIR, we embed the CPPCL into a coupled dictionary learning framework. Given the feature matrices $\mathbf{F}^{\mathrm{S}}$ and $\mathbf{F}^{\mathrm{I}}$ of the sketch and image domain, and two N-word dictionaries, one per modality: $\mathbf{D}^{\mathrm{S}} = [\mathbf{d}_n^{\mathrm{S}}]_{n=1}^{N} \in \mathbb{R}^{m_{\mathrm{S}} \times N}$ and $\mathbf{D}^{\mathrm{I}} = [\mathbf{d}_n^{\mathrm{I}}]_{n=1}^{N} \in \mathbb{R}^{m_{\mathrm{I}} \times N}$, we learn the associated dictionaries $\mathbf{D}^{\mathrm{S}}$, $\mathbf{D}^{\mathrm{I}}$ and sparse

representations $\mathbf{C}^{\text{S}}, \mathbf{C}^{\text{I}}$ by minimizing the following objective function:

$$\mathcal{L}_{\text{RL}} = \|(\mathbf{F}^{\text{S}} - \mathbf{D}^{\text{S}}\mathbf{C}^{\text{S}})\mathbf{V}^{\text{S}}\|_{\mathcal{F}}^2 + \|(\mathbf{F}^{\text{I}} - \mathbf{D}^{\text{I}}\mathbf{C}^{\text{I}})\mathbf{V}^{\text{I}}\|_{\mathcal{F}}^2 + \alpha\left(\|\mathbf{C}^{\text{S}}\|_1 + \|\mathbf{C}^{\text{I}}\|_1\right) + \beta \operatorname{Tr}\left(\mathbf{C}^{\text{J}}\mathbf{V}^{\text{J}}\mathbf{L}\mathbf{V}^{\text{J}\top}\mathbf{C}^{\text{J}\top}\right),$$

subject to:

$$\|\mathbf{d}_n^{\text{S}}\|, \|\mathbf{d}_n^{\text{I}}\| \leq 1 \quad \forall n, \qquad v_k^{\text{S}}, v_l^{\text{I}} \in \{0,1\} \quad \forall k,l,$$

where $\alpha \geq 0$ is a regularization parameter and $\|\cdot\|_{\mathcal{F}}$ denotes the Frobenius norm. The constraints remove any scale ambiguities due to the matrix products $\mathbf{D}^{\text{S}}\mathbf{C}^{\text{S}}$ and $\mathbf{D}^{\text{I}}\mathbf{C}^{\text{I}}$, while the regularization terms induce sparsity in the learned codes.

We also introduce a graph Laplacian regularizer to maintain the relational link between the learned representations of sketches and images in the training set. Ideally, each sketch corresponds to at least an image (*e.g.* for sketch to photo face recognition [155] in the context of security and biometrics applications). Alternatively, the association among sketches and images is derived from image class information [55]. Generally speaking, in this paper we consider both intra-modality and cross-modality relationships, modeled by a non-negative weight matrix $\mathbf{W} \in \mathbb{R}^{+(K+L)\times(K+L)}$. Intuitively, the larger w_{pq} is, the stronger the relationship between the p-th and q-th codes is. Importantly, when $1 \leq p,q \leq K$ (respectively, $K < p,q \leq K+L$), w_{pq} relates two sketches (respectively, two images) creating an intra-modality link, otherwise w_{pq} relates a sketch and an image (cross-modality link). Interpreting $\mathbf{W}$ as the weight matrix of a graph and denoting the associated Laplacian matrix[2] by $\mathbf{L}$, a graph laplacian regularizer for the codes is defined as $\operatorname{Tr}\left(\mathbf{C}^{\text{J}}\mathbf{L}\mathbf{C}^{\text{J}\top}\right) = \frac{1}{2}\sum_{p,q=1}^{K+L} w_{pq}\|\mathbf{c}_p^{\text{J}} - \mathbf{c}_q^{\text{J}}\|^2$, where $\mathbf{C}^{\text{J}} = [\mathbf{c}_p^{\text{J}}]_{p=1}^{K+L} = [\mathbf{C}^{\text{S}}\,\mathbf{C}^{\text{I}}] \in \mathbb{R}^{N\times(K+L)}$ is a joint code matrix, and $\beta \geq 0$ is a regularization parameter controlling the importance of the relational knowledge. By embedding pacing variables $\mathbf{V}^{\text{J}}$ into $\operatorname{Tr}\left(\mathbf{C}^{\text{J}}\mathbf{L}\mathbf{C}^{\text{J}\top}\right)$, we obtain the self-paced graph laplacian regularizer $\operatorname{Tr}\left(\mathbf{C}^{\text{J}}\mathbf{V}^{\text{J}}\mathbf{L}\mathbf{V}^{\text{J}\top}\mathbf{C}^{\text{J}\top}\right)$. Finally, the optimization problem to solve for writes:

$$\begin{aligned}
\min_{\mathbf{D}^{\text{S}},\mathbf{D}^{\text{I}},\mathbf{C}^{\text{J}},\mathbf{V}^{\text{J}},\boldsymbol{\xi}^{\text{J}}} \; & \mathcal{L}_{\text{RL}} + f_{\text{PC}}(\boldsymbol{\xi}^{\text{J}};\mathcal{U}^{\text{S}},\mathcal{U}^{\text{I}}) + f_{\text{SP}}(\mathbf{V}^{\text{J}},\gamma) \\
\text{s.t. } & \|\mathbf{d}_n^{\text{S}}\|, \|\mathbf{d}_n^{\text{I}}\| \leq 1 \quad \forall n, \\
& v_k^{\text{S}}, v_l^{\text{I}} \in \{0,1\} \quad \forall k,l, \\
& v_k^{\text{S}} - v_{k'}^{\text{S}} < \xi_{kk'}^{\text{S}}, \xi_{kk'}^{\text{S}} \geq 0, \forall (k,k') \in \mathcal{U}^{\text{S}} \\
& v_l^{\text{I}} - v_{l'}^{\text{I}} < \xi_{ll'}^{\text{I}}, \xi_{ll'}^{\text{I}} \geq 0, \forall (l,l') \in \mathcal{U}^{\text{I}}.
\end{aligned} \tag{2.5}$$

2.3.4 Laplacian and Curricula Construction

In this section we describe how we construct the modality-specific partial curricula and the Laplacian matrix representing the relational knowledge. However, it is worth noting that our approach is general and other design choices are possible. We build both the curricula and the Laplacian in the training set from the sketch and image features and a group association, that could arise from the class membership or from unsupervised clustering. In our experiments, we also devised a protocol to construct a curriculum for sketches from

[2]The Laplacian matrix of a graph with weight matrix $\mathbf{W}$ is defined as $\mathbf{L} = \mathbf{D} - \mathbf{W}$, where $\mathbf{D}$ is a diagonal matrix with $d_{pp} = \sum_q w_{pq}$.

human manual annotations.

Construction of graph Laplacian matrix

To build the Laplacian matrix (computed from the weights w_{pq}), the intra-modality relationships are defined using the Gaussian kernel and the inter-modality with group association, as in [181]:

$$w_{pq} = \begin{cases} \mathrm{e}^{-\left\|\mathbf{f}_p^{\mathrm{S}} - \mathbf{f}_q^{\mathrm{S}}\right\|_2^2 / 2\sigma^2}, & p, q \leq K \\ \mathrm{e}^{-\left\|\mathbf{f}_{p-K}^{\mathrm{I}} - \mathbf{f}_{q-K}^{\mathrm{I}}\right\|_2^2 / 2\sigma^2}, & K < p, q \\ 1, & \begin{array}{l} p \leq K < q \text{ and } p \sim q \\ q \leq K < p \text{ and } q \sim p \end{array} \\ 0, & \text{otherwise,} \end{cases} \tag{2.6}$$

where σ is the Gaussian kernel parameter fixed to 1 with no significant performance variation around this value. The symbol $\sim$ indicates samples belonging to the same cluster/class.

Construction of modality-specific curricula

Regarding the curricula construction, as stated above, a fundamental aspect of the the proposed framework is the possibility to handle partial curricula. Previous CL or hybrid CL-SPL methods [9, 62] instead assume that a full curriculum, *i.e.* a complete order of samples, is provided. This is a strong assumption that may be unrealistic in real-world large-scale tasks. On the one hand, even if automatic measures of the easiness of an image [80] have been developed, these metrics are accurate up to some extent and therefore deriving a full ranking from these measures may be inappropriate. On the other hand, manually annotating the entire set of images represents a huge human workload, highly demanding for medium and large-scale datasets. In addition, if the multi-modal dataset is gathered incrementally, the cost of updating the curriculum grows with the size of the dataset.

Image partial curricula. The partial curricula for the image domain is obtained by means of an automated procedure based on previous studies [80, 2]. Intuitively, easy images are those containing non-occluded high-resolution objects in low-cluttered background. Previous works [80] proposed to define the easiness of an image from the "objectness" measures [2]. In the same line of though, we compute the easiness measure as the median of the 30 highest "objectness" scores among a set of 1,000 window proposals. An example is shown in Figure 2.3. This procedure approximates the easiness of a training image. Notice that, two images with largely different scores are likely to correspond to samples with different easiness. On the contrary, if the scores are similar, imposing that the image with the lowest score is the easiest in the pair may induce some errors. The constraint associated to an image pair is included in $\mathcal{U}^{\mathrm{I}}$ only if the difference of their associated scores exceeds a certain threshold δ^{I} (*i.e.* if one of the images in the pair is significantly easier than the other).

Sketch partial curricula. Contrary to the image domain, there is no widely-accepted procedure to define the easiness of a sketch. Therefore, we consider two methods for constructing the partial curriculum in the sketch domain. The first one is an automatic method that follows again the philosophy of [2]. Given a sketch, we randomly sample 100 windows at different scales and positions. For each window we compute the "edgeness"

Figure 2.3: An illustration of objectness generation process for assessing the easiness of an image sample.

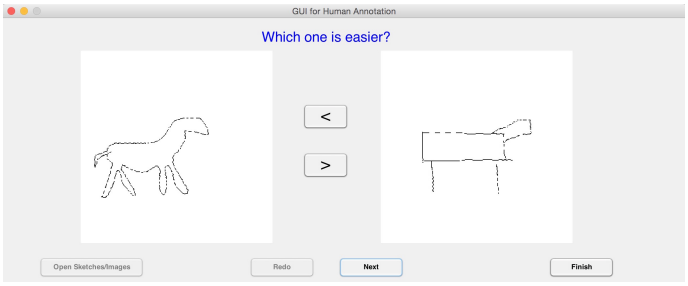


Figure 2.4: The graphical interface used for annotation. Easy sketches are those with more details and easy images are those with non-occluded high-resolution objects in low-cluttered background.

score, representing the edge density within the window as proposed in [2]. Intuitively, the edgeness should follow the rationale that easy sketches are those with more details. As previously done for images, the constraint associated to a pair of sketches is included into $\mathcal{U}^{\mathrm{S}}$ only if their measure of edgeness differs by at least δ^{S}. The second one is a semi-automatic strategy for building the partial curricula of the sketches by including human annotators in the loop. A naive retrieval method based on SHOG features [34] generates potential constraints (pairs of sketches). In details, we pair each sketch with the closest among the cluster/class. The human annotator is then queried which sketch is easier to learn from. Ten PhD students (6 male, 4 female) of age 24.3 ± 1.4 (mean, standard deviation) performed the annotation after being instructed that "easy" sketches meant sketches with more details. Importantly, since CPPCL is specifically designed to handle partial curricula, annotators had the possibility to "skip" sketch pairs if they were unable to decide. A simple GUI, shown in Figure 2.4, was developed for annotation.

2.4 Model Optimization

The optimization problem in Eqn. 2.5 is not jointly convex in all variables. However, efficient alternate optimization techniques can solve it since it is convex on $\{\mathbf{D}^{\mathrm{S}}, \mathbf{D}^{\mathrm{I}}\}$, $\{\mathbf{C}^{\mathrm{J}}\}$ and $\{\mathbf{V}^{\mathrm{J}}, \boldsymbol{\xi}^{\mathrm{J}}\}$ when the other two sets of variables are fixed. We proposed two different self-paced regularizers f_{SP_A} and f_{SP_B} in our model. However, they have no impact when solving for $\mathbf{D}^{\mathrm{S}}$ and $\mathbf{D}^{\mathrm{I}}$, while we provide two different solutions for solving $\mathbf{V}^{\mathrm{J}}$ and $\boldsymbol{\xi}^{\mathrm{J}}$.

Solve for $\mathbf{D}^{\mathrm{S}}$ and $\mathbf{D}^{\mathrm{I}}$

Fixing $\mathbf{C}^{\mathrm{J}}$, $\mathbf{V}^{\mathrm{J}}$ and $\boldsymbol{\xi}^{\mathrm{J}}$, the optimization problem for $\mathbf{D}^{\mathrm{S}}$ (analogously for $\mathbf{D}^{\mathrm{I}}$) writes:

$$\min_{\mathbf{D}^{\mathrm{S}}} \|(\mathbf{F}^{\mathrm{S}} - \mathbf{D}^{\mathrm{S}}\mathbf{C}^{\mathrm{S}})\mathbf{V}^{\mathrm{S}}\|_2^2 \quad \text{s.t.} \quad \|\mathbf{d}_k^{\mathrm{S}}\| \leq 1. \tag{2.7}$$

This problem is a Quadratically Constrained Quadratic Program (QCQP) that can be solved using gradient descent with e.g. Lagrangian duality [79].

Solve for $\mathbf{C}^{\mathrm{J}}$

By fixing $\mathbf{D}^{\mathrm{S}}$, $\mathbf{D}^{\mathrm{I}}$, $\mathbf{V}^{\mathrm{J}}$ and $\boldsymbol{\xi}^{\mathrm{J}}$ the optimization function for the codes can be rewritten as:

$$f(\mathbf{C}^{\mathrm{J}}) = \|(\mathbf{F}^{\mathrm{S}} - \mathbf{D}^{\mathrm{S}}\mathbf{C}^{\mathrm{S}})\,\mathbf{V}^{\mathrm{S}}\|_{\mathcal{F}}^2 + \|(\mathbf{F}^{\mathrm{I}} - \mathbf{D}^{\mathrm{I}}\mathbf{C}^{\mathrm{I}})\,\mathbf{V}^{\mathrm{I}}\|_{\mathcal{F}}^2 + \alpha\,\|\mathbf{C}^{\mathrm{J}}\|_1 + \beta\,\mathrm{Tr}\left(\mathbf{C}^{\mathrm{J}}\mathbf{V}^{\mathrm{J}}\mathbf{L}\mathbf{V}^{\mathrm{J}\top}\mathbf{C}^{\mathrm{J}\top}\right). \tag{2.8}$$

According to FISTA [6], f can be viewed as a proximal regularization problem, solved using the following recursion (over r):

$$\mathbf{C}_r^{\mathrm{J}} = \underset{\mathbf{C}^{\mathrm{J}}}{\mathrm{argmin}}\left\{\frac{\left\|\mathbf{C}^{\mathrm{J}} - \mathbf{C}_{r-1}^{\mathrm{J}} + t_r\nabla f(\mathbf{C}_{r-1}^{\mathrm{J}})\right\|_{\mathcal{F}}^2}{2t_r} + \alpha\|\mathbf{C}^{\mathrm{J}}\|_1\right\}, \tag{2.9}$$

where $t_r > 0$ is the step size and $\nabla f(\mathbf{C}^{\mathrm{J}}) = [\nabla f(\mathbf{C}^{\mathrm{S}})\ \nabla f(\mathbf{C}^{\mathrm{I}})]$ is the concatenation of the two gradients defined as:

$$\nabla f(\mathbf{C}^{\mathrm{S}}) = 2\mathbf{D}^{\mathrm{S}\top}(\mathbf{D}^{\mathrm{S}}\mathbf{C}^{\mathrm{S}} - \mathbf{F}^{\mathrm{S}})(\mathbf{V}^{\mathrm{S}})^2 + 2\beta\,(\mathbf{C}^{\mathrm{S}}\mathbf{V}^{\mathrm{S}}\mathbf{L}^{\mathrm{S}} + \mathbf{C}^{\mathrm{I}}\mathbf{V}^{\mathrm{I}}\mathbf{L}^{\mathrm{IS}})\,\mathbf{V}^{\mathrm{S}}, \tag{2.10}$$

where the sublaplacian matrices are taken from the Laplacian matrix as $\mathbf{L} = [\mathbf{L}^{\mathrm{S}}\,\mathbf{L}^{\mathrm{SI}}; \mathbf{L}^{\mathrm{IS}}\,\mathbf{L}^{\mathrm{I}}]$. The second gradient, $\nabla f(\mathbf{C}^{\mathrm{I}})$ is defined analogously to $\nabla f(\mathbf{C}^{\mathrm{S}})$. Moreover, (2.9) is a standard LASSO problem whose optimal solution can be found using the feature-sign search algorithm in [79].

Solve for $\mathbf{V}^{\mathrm{J}}$ and $\boldsymbol{\xi}^{\mathrm{J}}$ with the regularizer $f_{\mathbf{SP}_A}$ We fix $\mathbf{D}^{\mathrm{S}}$, $\mathbf{D}^{\mathrm{I}}$, $\mathbf{C}^{\mathrm{J}}$ to solve for $\boldsymbol{\xi}^{\mathrm{J}}$ and $\mathbf{V}^{\mathrm{J}}$, and the problem writes:

$$\begin{aligned}
&\min_{\mathbf{V}^{\mathrm{J}}} \mathcal{L}_{\mathrm{RL}} - \gamma\left(\sum_{k=1}^{K}\frac{1}{E_{g,k}^{\mathrm{S}}}v_k^{\mathrm{S}} + \sum_{l=1}^{L}\frac{1}{E_{g',l}^{\mathrm{I}}}v_l^{\mathrm{I}}\right),\\
&\qquad + \mu\left(\sum_{(k,k')\in\mathcal{U}^{\mathrm{S}}}\xi_{kk'}^{\mathrm{S}} + \sum_{(l,l')\in\mathcal{U}^{\mathrm{I}}}\xi_{ll'}^{\mathrm{I}}\right)\\
&\text{s.t. } 0 \le v_k^{\mathrm{S}}, v_l^{\mathrm{I}} \le 1 \quad \forall k,l,\\
&\qquad v_k^{\mathrm{S}} - v_{k'}^{\mathrm{S}} < \xi_{kk'}^{\mathrm{S}}, \xi_{kk'}^{\mathrm{S}} \ge 0, \forall (k,k') \in \mathcal{U}^{\mathrm{S}},\\
&\qquad v_l^{\mathrm{I}} - v_{l'}^{\mathrm{I}} < \xi_{ll'}^{\mathrm{I}}, \xi_{ll'}^{\mathrm{I}} \ge 0, \forall (l,l') \in \mathcal{U}^{\mathrm{I}}.
\end{aligned} \tag{2.11}$$

Here we replace the self-paced regularizer $\sum_{g=1}^{G^{\mathrm{S}}}\frac{1}{E_g^{\mathrm{S}}}\|\mathbf{V}^{\mathrm{S}}\mathbf{p}_g^{\mathrm{S}}\|_1$ with $\sum_{k=1}^{K}\sum_{g=1}^{G^{\mathrm{S}}}\frac{1}{E_g^{\mathrm{S}}}p_{g,k}^{\mathrm{S}}v_k^{\mathrm{S}} = \sum_{k=1}^{K}\frac{1}{E_{g,k}^{\mathrm{S}}}v_k^{\mathrm{S}}$ since $v_i \ge 0$, $E_{g,k}^{\mathrm{S}}$ being the size of group/class g of sample k. As discussed in Section 2.3.2 and following [187, 159], the self-paced regularizer can be used with continuous self-pacing variables for facilitating the optimization. This property is particularly advantageous in our case, because the joint $(\mathbf{V}^{\mathrm{J}}, \boldsymbol{\xi}^{\mathrm{J}})$ optimization problem can now be treated as a quadratic programming (QP) problem with a set of linear inequality constraints. Recent studies have shown that this strategy, as opposed to solving the original mixed integer quadratic programming problem, is successful in several applications [187, 62].

Let $\mathbf{y} = [[v_k^{\mathrm{S}}]_k[v_l^{\mathrm{I}}]_l[\xi_{kk'}^{\mathrm{S}}]_{kk'}[\xi_{ll'}^{\mathrm{I}}]_{ll'}] \in \mathbb{R}^{K+L+C^{\mathrm{S}}+C^{\mathrm{I}}}$ denote the joint optimization variable

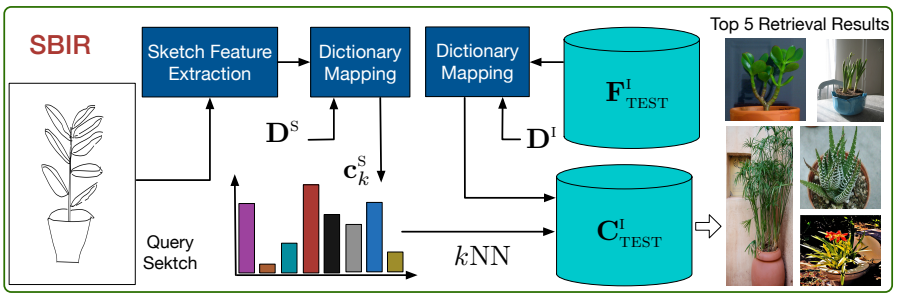


Figure 2.5: An illustration of the test phase of the proposed model for SBIR.

for which the problem writes:

$$\min_{\mathbf{y}} \mathbf{y}^\top \mathbf{R}\mathbf{y} + \mathbf{b}^\top \mathbf{y} \qquad (2.12)$$
$$\text{s.t. } \mathbf{G}\mathbf{y} \leq \mathbf{h},$$

where the values of $\mathbf{R}$, $\mathbf{b}$, $\mathbf{G}$ and $\mathbf{h}$ are defined in the following. $\mathbf{R}$ is a $(K+L+C^{\mathrm{S}}+C^{\mathrm{I}}) \times (K+L+C^{\mathrm{S}}+C^{\mathrm{I}})$ matrix with all zeros except for the first $(K+L)\times(K+L)$ block, where C^{S} and C^{I} denote the number of constraints of the sketch and the image modality respectively. More precisely:

$$R_{pq} = \begin{cases} \|\mathbf{f}^{\mathrm{S}}_p - \mathbf{D}^{\mathrm{S}}\mathbf{c}^{\mathrm{S}}_p\|^2 & q = p \leq K \\ \|\mathbf{f}^{\mathrm{I}}_{p-K} - \mathbf{D}^{\mathrm{I}}\mathbf{c}^{\mathrm{I}}_{p-K}\|^2 & K < q = p \leq L+K \\ \beta w_{pq}\|\mathbf{c}^{\mathrm{J}}_p - \mathbf{c}^{\mathrm{J}}_q\|^2 & 1 \leq p \neq q \leq K+L \\ 0 & \text{otherwise} \end{cases}$$

and $\mathbf{b} = [-\frac{\gamma}{E^{\mathrm{S}}_{g,1}}, ..., -\frac{\gamma}{E^{\mathrm{S}}_{g,K}}, -\frac{\gamma}{E^{\mathrm{I}}_{g',1}}, ..., -\frac{\gamma}{E^{\mathrm{I}}_{g',L}}, \mu\mathbf{1}^\top_{C^{\mathrm{S}}+C^{\mathrm{I}}}]^\top$, where $\mathbf{1}_C$ is a $C \times 1$ vector filled with ones. $\mathbf{G}$ and $\mathbf{h}$ represent the inequality and bound constraints in (2.5) and their derivation is straightforward. Since there are $2(K+L+C^{\mathrm{S}}+C^{\mathrm{I}})$ constraints, $\mathbf{G} \in \mathbb{R}^{2(K+L+C^{\mathrm{S}}+C^{\mathrm{I}})\times(K+L+C^{\mathrm{S}}+C^{\mathrm{I}})}$ and $\mathbf{h} \in \mathbb{R}^{2(K+L)+C^{\mathrm{S}}+C^{\mathrm{I}}}$.

Solve for $\mathbf{V}^{\mathrm{J}}$ and $\boldsymbol{\xi}^{\mathrm{J}}$ with the regularizer f_{SP_B} Similar to the previous case with the regularizer f_{SP_A}, by fixing the dictionaries $\mathbf{D}^{\mathrm{S}}$, $\mathbf{D}^{\mathrm{I}}$ and the codes $\mathbf{C}^{\mathrm{J}}$, the optimization problem is also a QP problem, and the only difference is that R_{pq} and $\mathbf{b}$ change. In this case, $\mathbf{b} = [\gamma\mathbf{1}^\top_{K+L}\ \mu\mathbf{1}^\top_{C^{\mathrm{S}}+C^{\mathrm{I}}}]^\top$ and R_{pq} becomes:

$$R_{pq} = \begin{cases} \|\mathbf{f}^{\mathrm{S}}_p - \mathbf{D}^{\mathrm{S}}\mathbf{c}^{\mathrm{S}}_p\|^2 - \gamma/2 & q = p \leq K \\ \|\mathbf{f}^{\mathrm{I}}_{p-K} - \mathbf{D}^{\mathrm{I}}\mathbf{c}^{\mathrm{I}}_{p-K}\|^2 - \gamma/2 & K < q = p \leq L+K \\ \beta w_{pq}\|\mathbf{c}^{\mathrm{J}}_p - \mathbf{c}^{\mathrm{J}}_q\|^2 & 1 \leq p \neq q \leq K+L \\ 0 & \text{otherwise} \end{cases}$$

Then the QP problem can be effectively solved with the interior-point algorithms [146]. The full optimization procedure is shown in Algorithm 1.

Test Phase for Sketch-to-Image Retrieval

Fig. 2.5 depicts the test phase of the proposed approach. Given the learned dictionaries $\mathbf{D}^{\mathrm{I}}$ and features of the retrieval images $\mathbf{F}^{\mathrm{I}}_{\mathrm{TEST}}$, we perform a dictionary mapping to calculate

Algorithm 1: Optimization Procedure

Input: the features $\mathbf{F}^{\mathrm{S}}$, $\mathbf{F}^{\mathrm{I}}$ and the parameters α, β, γ, μ
Output: $\mathbf{D}^{\mathrm{S}}$, $\mathbf{C}^{\mathrm{S}}$, $\mathbf{D}^{\mathrm{I}}$, $\mathbf{C}^{\mathrm{I}}$

```
Initialize D^S, C^S, D^I, C^I as described in Section 2.5 and initialize a step size η
(η > 1);
while not converged do
    Update V^J and ξ^J following (2.12);
    Update C^S, C^I with (2.9);
    Update D^S, D^I by solving (2.7);
    γ ← ηγ;
end
return D^S*, C^S*, D^I*, C^I*
```

all the sparse representations $\mathbf{C}^{\mathrm{I}}_{\mathrm{TEST}}$ of the retrieval sketches via solving:

$$\min_{\mathbf{C}^{\mathrm{I}}_{\mathrm{TEST}}} \|(\mathbf{F}^{\mathrm{I}}_{\mathrm{TEST}} - \mathbf{D}^{\mathrm{I}}\mathbf{C}^{\mathrm{I}}_{\mathrm{TEST}})\|^2_{\mathcal{F}} + \alpha\|\mathbf{C}^{\mathrm{I}}_{\mathrm{TEST}}\|_1 . \tag{2.13}$$

For a query sketch k, a corresponding sparse representation $\mathbf{c}^{\mathrm{S}}_k$ can be calculated by a similar dictionary mapping with $\mathbf{D}^{\mathrm{S}}$ as in Eqn. (2.13). Then we retrieve top K results from $\mathbf{C}^{\mathrm{I}}_{\mathrm{TEST}}$ using K Nearest Neighbor (K-NN), while for tests on sketch-to-face recognition, a Nearest Neighbor classifier is used.

2.5 Experiments

To evaluate the effectiveness of our approach for Cross-Paced Representation Learning (CPRL), we conduct extensive experiments on four publicly available datasets: the CUHK Face Sketch (CUFS) [155], the Flickr15k [55], the Queen Mary SBIR [84] and the TU-Berlin Extension [31] datasets.

2.5.1 Implementation Details

The experiments were run on a PC with a quad core (2.1 GHz) CPU, 64GB RAM and an Nvidia Tesla K40 GPU. The proposed SBIR approach is implemented in Matlab and partially in C++ (the most computationally expensive components). For representing sketches, we adopted the mid-level representation method named Learned KeyShapes (LKS) [122]. We used a C++ implementation for efficient extraction of LKS features and wrap it in a Matlab interface. For representing images, CNN features were used. Specifically, the Caffe reference network 'AlexNet' pre-trained on ImageNet was used to extract features from the sixth (the first fully connected) layer. In all our experiments and for all datasets, the value of the self-paced parameter was initialized to $\gamma = 1$ and increased by a factor $\eta = 1.3$ at each iteration (until all the training samples are selected). The dictionaries $\mathbf{D}^{\mathrm{S}}$ and $\mathbf{D}^{\mathrm{I}}$ were initialized with joint DL [174] when both features have the same dimension and with modality-independent DL otherwise.

2.5.2 Sketch-to-Face Recognition

Dataset. We first carried out experiments on sketch to face recognition using the **CUFS** dataset, a very popular benchmark which contains sketch-face photo pairs collected from 188 CUHK students. Figure 2.6 shows some examples of sketch-photo pairs. The recognition task is to extract the face photo corresponding to a given sketch as described in Section 8.

Figure 2.6: Examples of sketch and face photo pairs of CUFS dataset.

Table 2.1: Average recognition rate for all benchmarked methods on CUFS for sketch-to-photo face recognition.

Method	Recognition Rate
Tang & Wang [140]	81.0%
Partial Least Squares (PLS) [131]	93.6%
Biliner model [143]	94.2%
Canonical Correlation Analysis (CCA)	94.6%
Semi-coupled Dictionary Learning (SCDL) [154]	95.2%
Joint Dictionary Learning (JDL) [174]	95.4%
Coupled Dictionary Learning (CDL) [56]	97.4%
CPRL with f_{SP_B} ($\beta = \gamma = \mu = 0$)	96.8%
CPRL with f_{SP_B} ($\gamma = \mu = 0$)	97.2%
CPRL with f_{SP_A} ($\mu = 0$)	**98.2%**
CPRL with f_{SP_B} ($\mu = 0$)	**98.6**%

We evaluated the performance of our approach on CUFS and compared it to other cross-domain retrieval methods and previous DL approaches.

Settings. Following [140], in our experiments 88 sketch-photo image pairs were randomly selected for training the model, and the remaining 100 pairs were used for testing. To fairly compare with previous works [174, 56], in this preliminary experiment we did not consider the powerful LKS sketch features and CNN image features, but we only used raw pixels as feature representations for the two modalities. We compared the proposed approach with several baseline methods including: canonical correlation analysis (CCA), partial least squares (PLS) [131], bilinear model [143], semi-coupled dictionary learning (SCDL) [154], joint dictionary learning (JDL) [174] and coupled dictionary learning (CDL) [56]. For the bilinear model, we used 70 PLS bases and 50 eigenvectors (see [131]). For all DL-based approaches we set the dictionary size to 50. In all cases, the recognition was performed using the nearest neighbor on the newly learned sparse representation as in [131, 56]. We implemented and evaluated two variants of our method considering two different self-paced regularizers f_{SP_A} and f_{SP_B} as introduced in Section 2.3.2. Furthermore, we explicitly evaluated the importance of the relational knowledge (β) and of self-pacing (γ). Since for CUFS both sketches and face images are quite homogeneous (*i.e.* , sketches were drawn by experts, faces in images are centered and equally illuminated), we did not use any curriculum by setting $\mu = 0$. The parameters α, β were set by cross-validation to 1 and 5, respectively.

Results. Table 2.1 shows the results of average recognition rate over five trials. CPRL with self-paced regularizer f_{SP_B} ($\mu = 0$) achieves the best average recognition rate: 98.6% (the influence of different self-paced regularizers is further analyzed in Section 2.5.4).

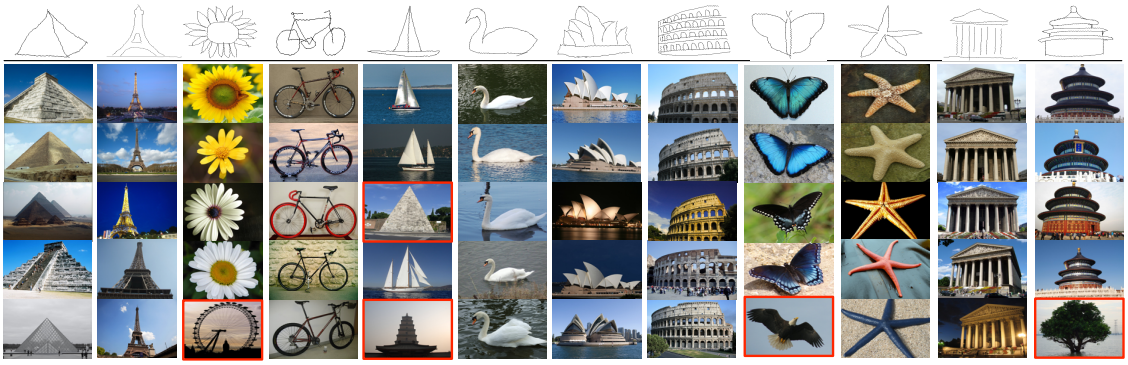
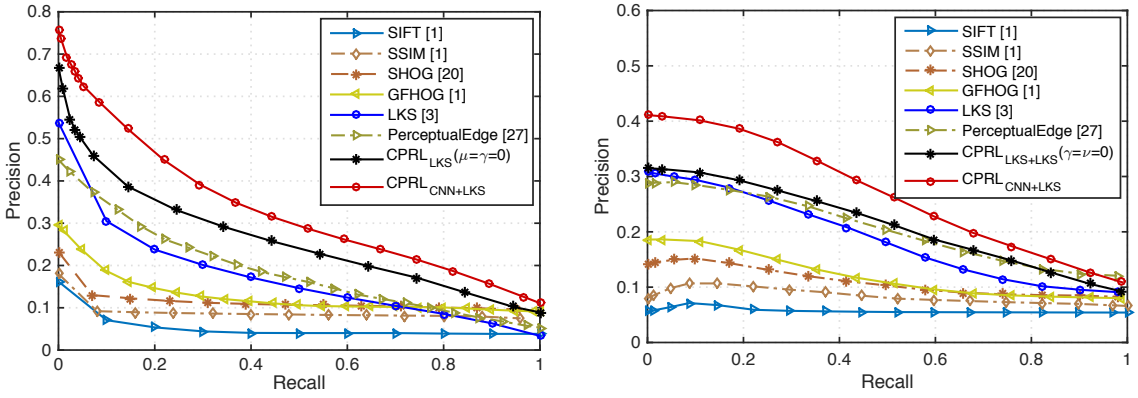


Figure 2.7: Top 5 retrieval results with sample query sketches in Flickr15K dataset. Red boxes show false positive retrievals.



(a) Flickr15K (b) QueenMary SBIR

Figure 2.8: Precision-Recall (PR) curves for the retrieval performance comparison of the different methods on Flirckr15K and QueenMary SBIR datasets.

Remarkably, CPRL with $f_{\mathrm{SP}_B}(\mu = 0)$ outperforms CDL, which is the best of the DL-based approaches, showing the advantage of using our self-paced scheme for learning robust cross-domain representations. Importantly, by setting the parameter β to 0, we notice that the effect of the relational knowledge is crucial in the performance of the overall method (CDL also uses relational knowledge). Among the compared methods, SCDL, JDL and CDL are the strongest competitors, achieving 95.2%, 95.4% and 97.4% recognition rate respectively. This means that DL is an effective strategy for learning cross-domain representations for the retrieval task. We also remark that CPRL with $f_{\mathrm{SP}_B}(\gamma = 0)$ outperforms the other two versions of CPRL, suggesting that the relational knowledge within the SP learning framework is beneficial for accurate retrieval.

2.5.3 Sketch-to-Image Retrieval

Datasets. We further performed the evaluation of CPRL on the Flickr15k and QueenMary SBIR datasets. The **Flickr15k dataset** is a widely used dataset for SBIR, containing around 14,660 images collected from Flickr and 330 free-hand sketches drawn by 10 non-expert sketchers. The dataset consists of 33 object categories and each sample is labeled with an object-class annotation. Since this dataset does not provide a training set, to evaluate our approach, we partitioned the dataset into a training set with randomly chosen 40% samples and a test set with the remaining samples. All the baseline methods were tested using the same setting for a fair comparison.

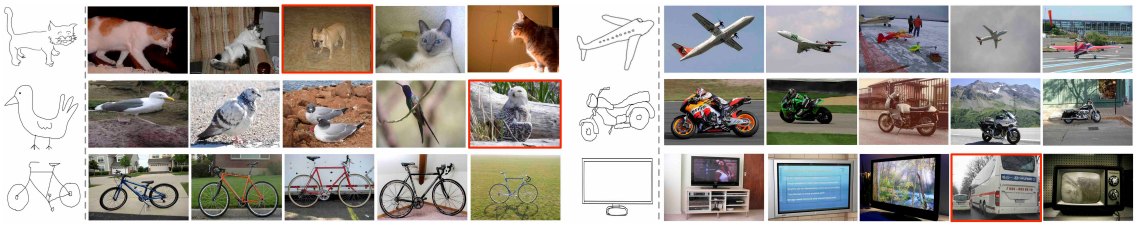


Figure 2.9: Top 5 retrieved images (right) using the query sketch samples (left) in the QueenMary SBIR dataset. Red boxes show false positive retrievals.

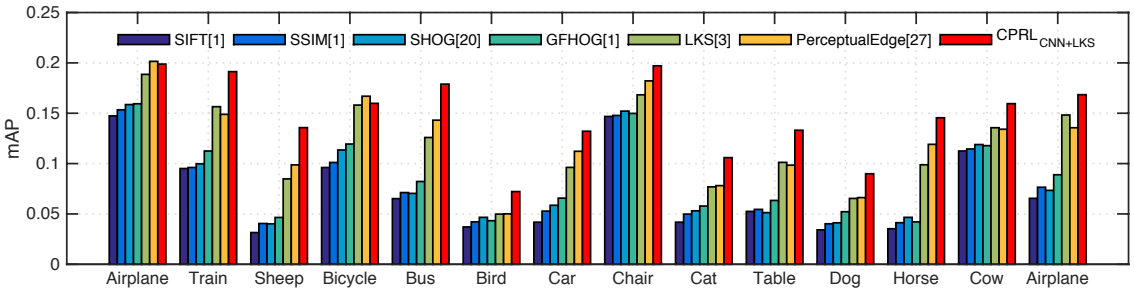


Figure 2.10: Retrieval performance comparison for each category of the Queen Mary SBIR dataset.

The **Queen Mary SBIR dataset** [84] is constructed by intersecting 14 common categories from the Eitz 20,000 sketch dataset [32] and the PASCAL VOC 2010 dataset [35], which consists of 1,120 sketches and 7,267 images. This dataset presents more complex conditions than the Flickr15k due to cluttered background and significant scale variations in the images. We use the official training and testing sets for evaluation. Since this dataset was originally used for fine-grained SBIR, while our task focuses on category-level SBIR, we only used image-level category annotations.

The **TU-Berlin Extension dataset** [31] consists of 250 object categories and each category has 80 free-hand sketches. Similar to [91], 204,489 extended natural images provided by [182] are added to TU-Berlin image gallery for the retrieval task.

Settings. To demonstrate the retrieval performance of CPRL, we compared with several state of the art SBIR methods, including SHOG [34], SIFT, SSIM, GFHOG evaluated in [55], Structure Tensor [33], Learned Key Shapes as in [122], PerceptualEdge [114], Sketch-a-Net (SaN) [179], Siamese CNN [115], GN Triplet [124], 3D shape [150] and DSH [91]. The first five methods first extract low-level feature representations (SHOG, SIFT, SSIM, GFHOG and StructureTensor) from the Canny edge maps of the images and the sketches respectively, and then generate the corresponding mid-level representations using a bag-of-words approach. Since the Queen Mary SBIR is a more difficult dataset than Flickr15, we considered SP-SHOG and SP-GFHOG instead of SHOG and GFHOG. Indeed, SP-HOG and SP-GFHOG employ a spatial pyramid model over SHOG and GFHOG features which has been demonstrated to provide more robust image representations than BoW [78]. LKS [122] learns mid-level sketch patterns named keyshapes. The learned keyshapes are used to construct image and sketch descriptors. PerceptualEdge [114] uses an edge grouping framework to create synthesized sketches from images. The retrieval is performed by querying the synthesized sketches instead of the images directly. Sketch-a-Net (SaN) [179] is an approach based on recent CNN architectures. Siamese CNN [115] uses a Siamese-based network structure for learning the similarity between the image and the

Table 2.2: Comparison of different methods on Flickr15k and Queen Mary SBIR datasets

Method	mAP	
	Flick15k	QueenMary SBIR
StructureTensor [33]	0.0801	0.0601
SIFT [55]	0.0967	0.0685
SSIM [55]	0.1068	0.0745
SHOG [34]	0.1152	0.0804
GFHOG [55]	0.1245	0.0858
LKS [122]	0.1640	0.1182
PerceptualEdge [114]	0.1741	0.1246
SaN [179]	0.1730	0.1211
Siamese CNN [115]	0.1954	-
$\text{CPRL}_{\text{GFHOG}}$ with f_{SP_B}	0.1693	0.1103
CPRL_{LKS} with f_{SP_B} $(\gamma = \mu = 0)$	0.2278	0.1265
CPRL_{LKS} with f_{SP_B}	0.2495	0.1467
$\text{CPRL}_{\text{CNN+LKS}}$ with f_{SP_A}	**0.2659**	**0.1521**
$\text{CPRL}_{\text{CNN+LKS}}$ with f_{SP_B}	**0.2734**	**0.1603**

sketch samples. DSH [91] jointly learns a hash function with the front-end CNN. For LKS and PerceptualEdge, we use the original codes provided by the authors with the same parameter setting described in the associated papers and we reimplemented other baselines whose codes are not publicly available. All the methods are evaluated on the same training/testing set for a fair comparison. If the original paper uses the same train/test split, the results are those reported in the paper.

To evaluate the proposed CPRL, we considered several settings using the self-paced regularizer f_{SP_B}: (i) CPRL_{LKS} $(\gamma = \mu = 0)$: CPRL without the curricula and self-pacing, using LKS features for both image and sketch domains; (ii) CPRL_{LKS}: CPRL using LKS features for both the image and the sketch modalities; (iii) $\text{CPRL}_{\text{CNN+LKS}}$: CPRL using CNN features for the image domain (*i.e.* features extracted from the sixth layer of the Caffe reference network trained on ImageNet) and LKS features for the sketch domain. We further considered the last baseline method with self-paced regularizer f_{SP_A}. The sketch curriculum, when used, is constructed using 60% of human annotations, since we did not observe any significant differences between the automatic and the manual procedures (see Section 2.5.4). For all CPRL methods, we set α, β, γ and N with cross-validation, and obtained $\alpha = 2$, $\beta = 25$, $\gamma = 0.5$ and $N = 1000$ for Flickr15k, and $\alpha = 6$, $\beta = 8$, $\gamma = 1$ and $N = 1500$ for Queen Mary SBIR.

Results. A performance comparison of different methods on the Flickr15k and the QueenMary SBIR datasets is shown in Table 2.2, reporting the mean average precision (mAP), and in Figure 2.8, depicting the precision-recall (PR) curve. Analyzing results on the Flickr15K dataset, three observations can be made: (i) $\text{CPRL}_{\text{CNN+LKS}}$ achieves the best mAP, showing a significant performance improvement (9.93 points) compared to the best state of the art method (0.1741 of PerceptualEdge [114]); (ii) $\text{CPRL}_{\text{GFHOG}}$ and CPRL_{LKS} compares favorably to GFHOG [55] and LKS [122] with 4.48 and 8.55 points improvement respectively, meaning that the advantage of the academic learning paradigm and its instantiation under the framework of dictionary learning is clear and independent of the features used; (iii) The clear performance gap when CPRL_{LKS} is compared to CPRL_{LKS} $(\gamma = \mu = 0)$ demonstrates the effectiveness of the proposed CPPCL strategy.

The fact that the best performance is obtained with $\text{CPRL}_{\text{CNN+LKS}}$ confirms our

Table 2.3: Comparison of different methods on the TU-Berlin Extension dataset.

Method	Feature Dimension	mAP
SHOG [34]	1296	0.091
GFHOG [55]	3500	0.119
SHELO [121]	1296	0.123
LKS [122]	1350	0.157
SaN [179]	512	0.154
Siamese CNN [115]	64	0.322
GN Triplet [124]	1024	0.187
3D shape [150]	64	0.054
DSH [91]	32 (bits)	**0.358**
DSH [91]	128 (bits)	**0.570**
CPRL_{LKS} with f_{SP_B} ($\gamma = \mu = 0$)	1000	0.269
CPRL_{LKS} with f_{SP_B}	1000	0.301
$\text{CPRL}_{\text{CNN+LKS}}$ with f_{SP_A}	1000	0.324
$\text{CPRL}_{\text{CNN+LKS}}$ with f_{SP_B}	1000	0.332

original intuition that different features can represent better the two different modalities. Interestingly the results in the table also show that our approach outperforms the SaN method [179] and Siamese CNN method [115], demonstrating the effectiveness of our framework in comparison with deep learning architectures. Finally, Figure 2.7 shows some qualitative results (top-five retrieved images) associated with the proposed method.

On the Queen Mary SBIR dataset, Table 2.2, $\text{CPRL}_{\text{CNN+LKS}}$ achieves an mAP of 0.1603 which is 3.57 points better than the best of all the comparison methods. It should be noted that this is not a trivial improvement on this very challenging dataset. $\text{CPRL}_{\text{CNN+LKS}}$ also outperforms CPRL_{LKS}, demonstrating the effectiveness of using different descriptors for sketches and images in SBIR. We also believe that LKS features are not robust enough to represent objects with various poses and cluttered background, as in Queen Mary SBIR dataset. CPRL_{LKS} obtains a clear improvement over CPRL_{LKS} ($\gamma = \mu = 0$), further verifying the usefulness of the proposed CPPCL scheme. Additionally, we show the retrieval performance for the category-level retrieval task in Figure 2.10. It is clear that for most of the classes (except for Airplane and Bicycle), $\text{CPRL}_{\text{CNN+LKS}}$ significantly outperforms all the comparison methods. Finally, Figure 2.9 reports the top 5 retrieval results of $\text{CPRL}_{\text{CNN+LKS}}$ for 10 query samples of sketches.

We further verify our performance on a larger SBIR dataset TU-Berlin Extension. The results are shown in Table 2.3. It is clear that CPRL_{LKS} with f_{SP_B} is significantly better than the LKS method and CPRL_{KS} with $f_{\text{SP}_B}(\gamma = \mu = 0)$, demonstrating the effectiveness of the proposed learning strategy. When using powerful CNNs features as input, our method obtains better performance than previous end-to-end trainable deep learning models [179, 115, 124, 150]. The DSH method in [91] achieves the best performance by successfully combining deep networks with hashing. We believe that our learning strategy is complementary to their method and the idea of exploiting curriculum and self-paced learning in the context of deep hashing is an interesting direction for future works.

2.5.4 In-depth Analysis of CPRL

In this section, we show the results of a further analysis of the proposed CPRL model on both the Flickr15k and the Queen Mary SBIR datasets. The analysis was conducted considering several aspects including sensitivity study, convergence analysis, effect of self-paced regularizers, impact of the curriculum construction and computational cost

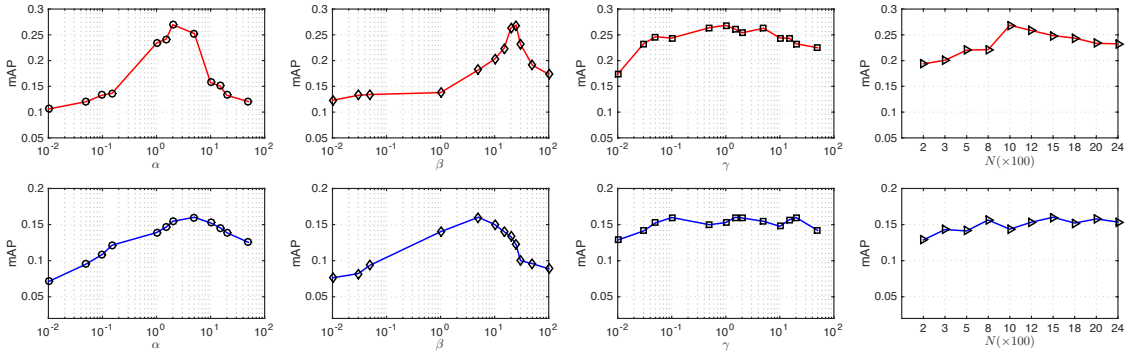


Figure 2.11: Empirical analysis of the model parameters: α, β, γ and the dictionary size N on Flickr15k (first row) and Queen Mary SBIR (second row) datasets.

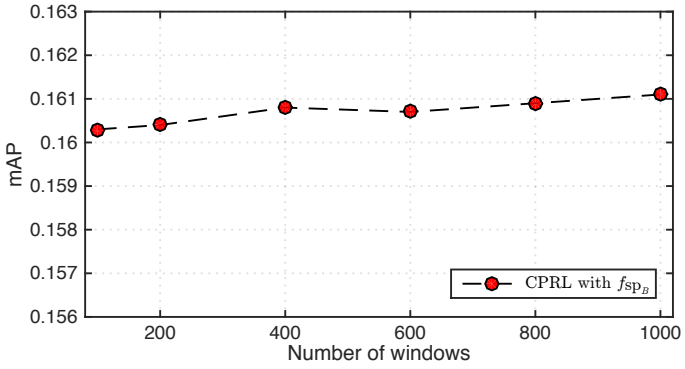


Figure 2.12: mAP at varying number of windows for edgeness calculation on the Queen Mary SBIR dataset.

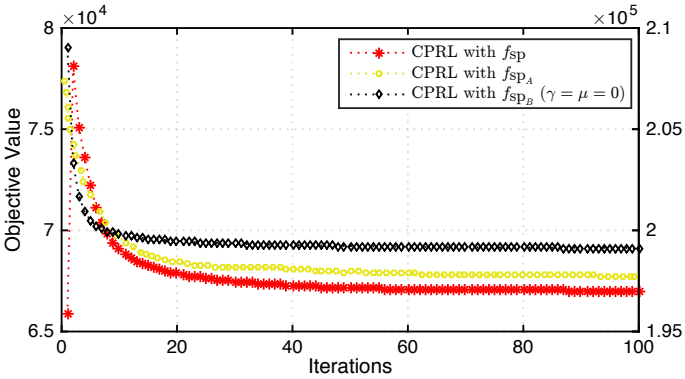


Figure 2.13: Convergence in terms of objective value for: CPRL with f_{SP_A} and with f_{SP_B} and CPRL with f_{SP_B} $(\gamma = \mu = 0)$.

analysis.

Sensitivity analysis. First, we assess the influence in performance of the different model parameters in CPRL. Figure 2.11 shows the mAP as a function of the parameters α, β, γ, N on both the Flickr15k and the Queen Mary SBIR datasets. The analysis on α, β and γ is in the range $[10^{-2}, 10^{2}]$, on N in the range $[200, 2400]$. It is clear from the plots that, while the method is sensitive to α and β, its retrieval performance does not change drastically within a wide range of γ and N. The sensitivity on β was already observed in previous research works [56]. The performance trend varying the different parameters shows some similarity on both datasets. We also conduct an analysis to evaluate the

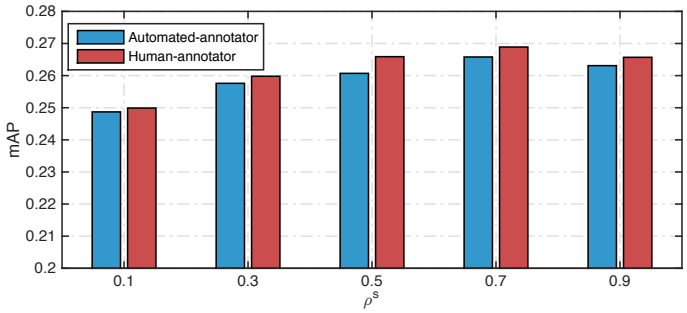


Figure 2.14: Performance considering automatically obtained and human-annotated sketch partial curricula.

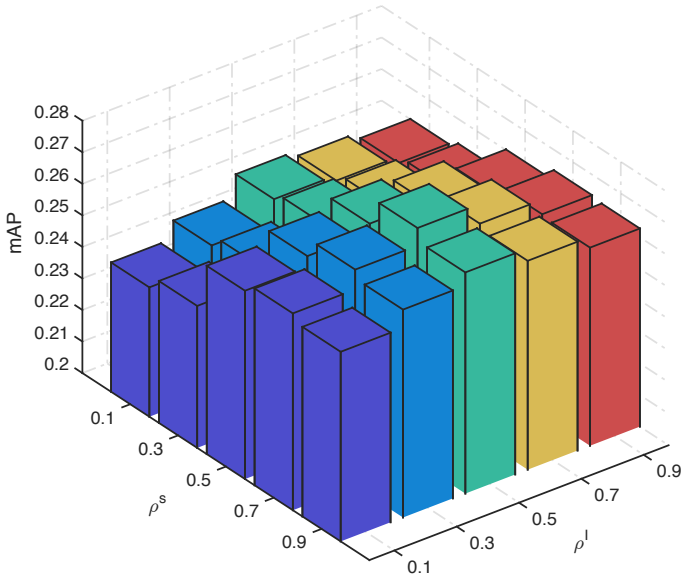


Figure 2.15: Performance at varying the constraints ratio of the sketch and the image modalities.

impact on the performance of the number of windows used for the edgeness calculation in the sketch domain (Fig. 2.12). Fig. 2.12 shows that the retrieval performance only slightly improves when increasing the number of windows. However, a large number of windows leads to a significant increase in terms of computational overhead. Therefore, we set the number of windows equal to 100 in our experiments as it represents a good trade-off between accuracy and computational cost.

Convergence analysis. Figure 2.13 plots the objective function value as a function of the iteration number for the proposed CPRL on Flickr15K with three different settings: (i) CPRL with f_{SP_A}; (ii) CPRL with f_{SP_B} and (iii) CPRL with f_{SP_B} ($\gamma = \mu = 0$). The results clearly show the convergence of the proposed iterative optimization procedure. All the three settings of CPRL attain a stable solution within less than 40 iterations, proving the efficiency of the algorithm proposed to solve the CPRL optimization problem. It is worth noting that both CPRL with f_{SP_A} and with f_{SP_B} obtain a much lower local minima than CPRL ($\gamma = \mu = 0$) (*e.g.* with f_{SP_B} giving 6.8×10^4 vs. 1.98×10^5), verifying the beneficial effect of the proposed CPPCL strategy for better optimization.

Analysis of self-paced regularizers. We carried out the retrieval experiments for CPRL with two different self-paced regularizers f_{SP_A} and f_{SP_B} on the four datasets. Table 2.1 and Table 2.2 show the quantitative results of the two CPRL variants. We

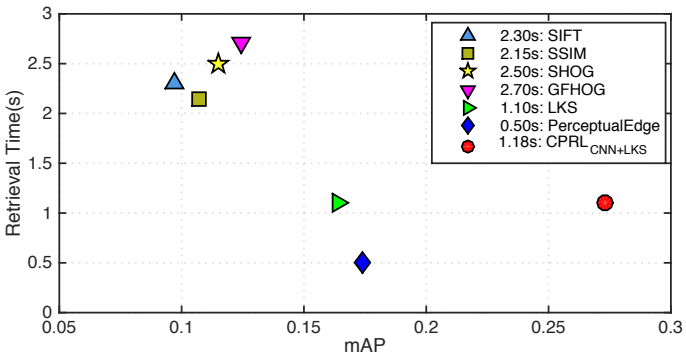


Figure 2.16: Comparison of the average retrieval time of different methods with respect to the mAP.

can observe that CPRL with f_{SP_B} slightly outperforms CPRL with f_{SP_A} on all the four datasets. We believe that this is probably due to the fact that when we optimize CPRL with f_{SP_B}, the self-pacing variables $\mathbf{V}$ are relaxed considering a real valued range [0, 1] (*i.e.* using a soft-weighting scheme) instead of discrete values. The soft weighting scheme is more effective than the hard weighting one in reflecting the true importance of samples in the training phase. This effect was previously observed in [187, 159].

Analysis of curriculum construction. To investigate the influence the modalitiy-specific curricula to the final retrieval performance, we plot the mAP as a function of ρ^{I} and ρ^{S}, the proportion of constraints used for the image and sketch curriculum relative to the number of possible constraints. Figure 2.15 shows the plot with ρ^{I} and ρ^{S} taking five values ranging from 0.1 to 0.9. We can observe that for both modalities, the use of the curricula indeed helps boosting the performance, while using the excess of constraints leads to a slight decrease in performance. This experimental finding supports our motivation of designing partial curricula learning in CPRL for SBIR. Our CPCL approach allows the human and the automated annotator to construct the partial curricula. To evaluate the difference of these two, Figure 2.14 plots the mAP as a function of ρ^{S} with ρ^{I} fixed to be 0.3. It is clear that the human annotations correspond to more effective partial curricula, but yet the difference when compared with the automated curricula constructions is small.

Computational cost analysis. In the following, we analyze the computational time overhead on Flickr15K experiments both in the off-line training phase and during the online retrieval phase. The training phase of our method mainly contains three steps: (i) feature extraction, (ii) curriculum construction and (iii) CPRL optimization. The input for CPRL are CNN features (for images) with size 4883×2400 and LKS features (for sketches) with size 132×2400, where 4833 and 132 are the number of training image and sketch samples respectively, and 2400 is the feature dimension. Table 2.4 reports computational times of different steps of the method. For the feature extraction, we consider CNN features from the image domain, which cost around 0.04 seconds per image sample. The CNN image features were extracted with the GPU. LKS is used to extract features from the sketch domain. The automated curriculum construction takes around 8 minutes and training CPRL and CPRL ($\gamma = \mu = 0$) with 50 iterations costs 27 and 21 minutes, respectively.

Online retrieval efficiency is a very important performance index for SBIR, especially for large-scale retrieval scenarios. Figure 2.16 plots the online retrieval time with respect to the mAP and compares CPRL with the state-of-the-art SBIR methods. Our $\text{CPRL}_{\text{CNN+LKS}}$ is based on three steps for the retrieval: (i) feature extraction from a query sketch sample

Table 2.4: Computational cost of the different training steps.

Phase	Component	Time overhead
Feature Extraction	CNN (for images)	0.04 ± 0.01 sec/sample
	LKS (for sketches)	1.1 ± 0.02 sec/sample
Training	Curriculum Construction	8 ± 1 min
	CPRL	27 ± 2 min
	CPRL($\gamma = \mu = 0$)	21± 2 min
Retrieval	CPRL	1.1 ± 0.1 sec/sample

using LKS, (ii) dictionary mapping to obtain a new feature representation and (iii) query the image features database with k-NN. The last two steps are very fast, and the feature extraction using LKS takes around 1 second. The average retrieval time for each query sample is around 1.18 seconds. PerceptualEdge method achieves the best retrieval speed, as it uses only two steps namely the HOG feature extraction and direct matching. The retrieval speed of ours is comparable to the LKS method, and is almost 2 times faster than GFHOG, SHOG, SIFT and SSIM, which first extract features, and then construct bag-of-words descriptors and finally perform the retrieval. The reason is that the step of constructing the bag-of-words features is more time consuming than the dictionary mapping step. More importantly, our approach obtains a very good balance between the retrieval performance (mAP) and the computational efficiency.

To conclude, our approach achieves better or comparable speed than previous works based on direct feature matching. We believe that other strategies can be used to further speed up the retrieval process, such as adopting hash-based algorithms. While this is not the focus of the current paper, our framework can be also extended in this direction.

2.6 Conclusion

We presented a novel cross-domain representation learning framework for computing robust cross-modal features for sketch-based image retrieval. In particular, this work explores self-paced and curriculum learning schemes for dictionary learning. A novel cross-paced partial curriculum learning strategy is designed to learn from samples with an easy-to-hard order, such as to avoid bad local optimal into dictionary learning optimization. The proposed framework naturally handles different descriptors for the sketch and the image domains. Therefore, domain-specific discriminative feature representations (*e.g.* , CNN features for images) are considered, overcoming the limitations of previous works. Extensive evaluation on four publicly available datasets shows that our approach achieves very competitive performance over state-of-the art methods for SBIR.

In this work CPPCL is instantiated within a coupled dictionary learning model for addressing the SBIR task. However, CPPCL is a general strategy which can be also combined with other representation learning methods. Future works will explore the adoption of CPPCL into cross-domain deep learning models [165].

CHAPTER 3

Learning Cross-Modal Deep Representations for Robust Pedestrian Detection [1]

This chapter presents a novel method for detecting pedestrians under adverse illumination conditions. Our approach relies on a novel cross-modality learning framework and it is based on two main phases. First, given a multimodal dataset, a deep convolutional network is employed to learn a non-linear mapping, modeling the relations between RGB and thermal data. Then, the learned feature representations are transferred to a second deep network, which receives as input an RGB image and outputs the detection results. In this way, features which are both discriminative and robust to bad illumination conditions are learned. Importantly, at test time, only the second pipeline is considered and no thermal data are required. Our extensive evaluation demonstrates that the proposed approach outperforms the state-of-the-art on the challenging KAIST multispectral pedestrian dataset and it is competitive with previous methods on the popular Caltech dataset.

3.1 Introduction

Great strides in pedestrian detection research [8] have been made for challenging situations, such as cluttered background, substantial occlusions and tiny target appearance. As for many other computer vision tasks, in the last few years significant performance gains have been achieved thanks to approaches based on deep networks [107, 3, 83, 145]. Additionally, the adoption of novel sensors, *e.g.* thermal and depth cameras, has provided new opportunities, advancing the state-of-the-art on pedestrian detection by tackling problems such as adverse illumination conditions and occlusions [57, 42, 113]. However, the vast majority of wide camera networks in surveillance systems still employ traditional RGB sensors and detecting pedestrians in case of illumination variation, shadows, and low external light is still a challenging open issue.

This paper introduces a novel approach based on Convolutional Neural Networks (CNN) to address this problem. Our method is inspired by recent works demonstrating that learning deep representations from cross-modal data is greatly beneficial for detection

[1] Dan Xu, Wanli Ouyang, Elisa Ricci, Xiaogang Wang, Nicu Sebe, "Learning Cross-Modal Deep Representations for Robust Pedestrian Detection", IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017).

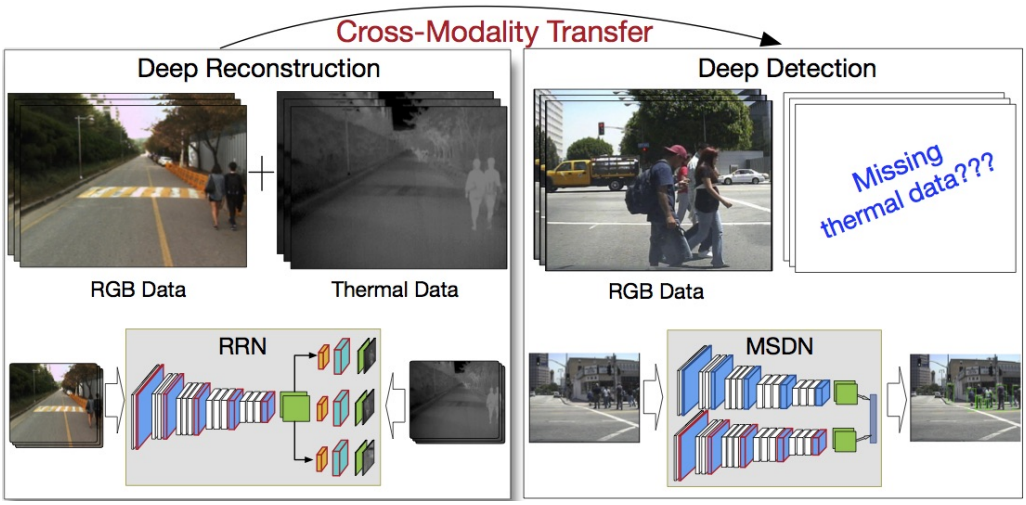


Figure 3.1: Overview of our framework. Our approach relies on two networks. The first network, named Region Reconstruction Network (RRN) is used to learn a non-linear feature mapping between RGB and thermal image pairs. Then, the learned model is transferred to a target domain where thermal inputs are no longer available and a second network, the Multi-Scale Detection Network (MDN), is used for learning an RGB-based pedestrian detector.

and recognition tasks [46, 51]. However, most approaches assume the availability of large annotated datasets. In the specific case of pedestrian detection, the community can rely on a great abundance of visual data gathered with surveillance cameras, cars and robotic platforms, but there are few labeled multi-modal datasets. Therefore, motivated by the successes of recent unsupervised deep learning techniques, we introduce an approach for learning cross-modal representations for pedestrian detection which does not require pedestrian bounding box annotations. More specifically, we propose leveraging information from multispectral data and using a deep convolutional network to learn a non-linear mapping from RGB to thermal images without human supervision. This cross-modal mapping is then exploited by integrating the learned representations into a second deep architecture, operating on RGB data and effectively modeling multi-scale information. Importantly, at test time, thermal data are not needed and pedestrian detection is performed only on color images.

Figure 3.1 depicts an overview of the proposed approach. Our intuition, illustrated in Fig.3.2, is that, by exploiting multispectral data with the proposed method, it is easier to distinguish hard negative samples in color images (*e.g.* , electric poles or trees with appearance similar to pedestrians), thus improving the detection accuracy. Experimental results on publicly available datasets, where several frames are captured under bad illumination conditions, demonstrate the advantages of our approach over previous methods. To summarize the main contributions of this work are:

- We introduce a novel approach for learning and transferring cross-modal feature representations for pedestrian detection. With the proposed framework, data from the auxiliary modality (*i.e.* thermal data) are used as a form of supervision for learning CNN features from RGB images. There are two fundamental advantages in our strategy. First, multispectral data are not employed at the test phase. This is crucial when deploying robotics and surveillance systems, as only traditional cameras are needed, significantly decreasing costs. Second, no pedestrian annotations are required in the thermal domain.

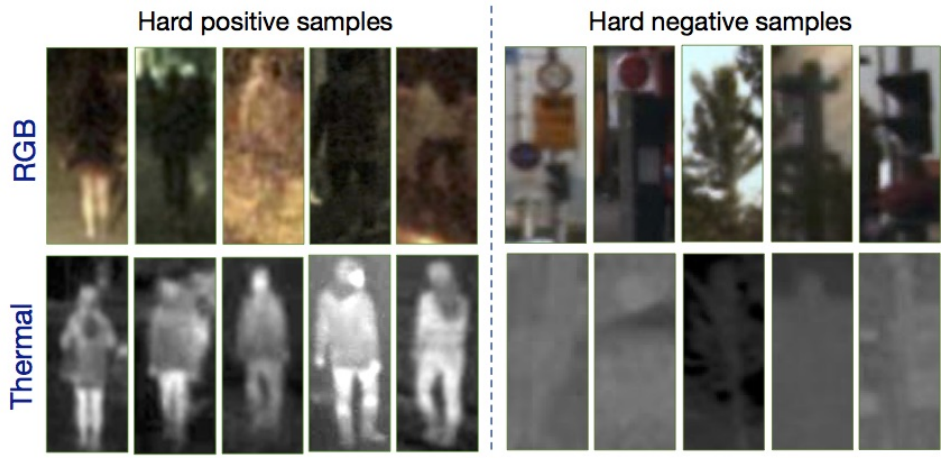


Figure 3.2: Motivation of this work. By exploiting thermal data in addition to RGB samples, it is easier to discriminate among pedestrians and background clutter.

This greatly reduces human labeling efforts and permits to exploit large data collections of RGB-thermal image pairs.

- To our knowledge, this is the first work specifically addressing the problem of pedestrian detection under adverse illumination conditions with convolutional neural networks. Previous works mostly adopted hand-crafted descriptors and integrated the thermal modality by using additional input features [57, 134]. Our approach is based on two novel deep network architectures, specifically designed for unsupervised cross-modal feature learning and for effectively transferring the learned representations.

- Through an extensive experimental evaluation, we demonstrate that our framework outperforms the state-of-the-art on the novel KAIST multispectral pedestrian dataset [57] and it is competitive with previous methods on the popular Caltech dataset [28].

This paper is organized as follows. Section 3.2 outlines related work on pedestrian detection and cross-modal feature learning. Section 3.3 describes the proposed framework for learning features robust to illumination variations in the context of pedestrian detection. Experimental results to demonstrate the benefits of our approach are presented in Section 3.4. We conclude with key remarks in Section 3.5.

3.2 Related Work

Research topics closely related to this work are pedestrian detection from surveillance videos and deep learning approaches operating on multimodal data. Below, we present a review of the most recent works on these topics.

3.2.1 Pedestrian Detection

Due to its relevance in many fields, such as robotics and video surveillance, the problem of pedestrian detection has received considerable interests in the research community. Over the years, a large variety of features and algorithms have been proposed for improving detection systems, both with respect to speed [148, 7, 3, 83] and accuracy [171, 108, 184, 185, 40, 145].

Recently, notable performance gains have been achieved with the adoption of powerful deep networks [107, 3], thanks to their ability to learn discriminative features directly from raw pixels. In [130], a CNN pre-trained with an unsupervised method based on

convolutional sparse coding was presented. The occlusion problem was addressed in [105], where a deep belief net was employed to learn the visibility masks for different body parts. This work was extended in [106] to model relations among multiple targets. More recently, in [144] DeepParts, a robust framework for handling severe occlusions, was presented. Differently from previous deep learning models addressing the occlusion problem, DeepParts does not rely on a single detector but it is based on multiple part detectors. Tian *et al.*[145] learned discriminative representations for pedestrian detection by considering semantic attributes of people and scenes. Cai *et al.*[14] introduced Complexity-Aware Cascade Training (CompACT), successfully integrating many heterogeneous features, both hand crafted and derived from CNNs. Zhang *et al.*[183] presented an approach based on the Region Proposal Network (RPN) [117] and boosted forests.

Other works focused on improving the computational times of CNN-based pedestrian detectors. For instance, Angelova *et al.*[3] proposed the DeepCascade method, *i.e.* a cascade of deep neural networks, and demonstrated a considerable gain in terms of detection speed. An in-depth analysis of different deep networks architectural choices for pedestrian detection was provided in [54]. To our knowledge, none of these previous works considers multi-modal data or tackles the problem of pedestrian detection under adverse illumination conditions.

Previous works have considered transferring information from other domains for constructing scene-specific pedestrian detectors. Wang *et al.*[152] proposed an unsupervised approach where target samples are collected by exploiting contextual cues, such as motions and scene geometry. Then, a pedestrian detector is built by re-weighting labeled source samples, *i.e.* by assigning more importance to samples more similar to target data. This approach was later extended in [180] to learn deep feature representations. Similarly, in [15] a sample selection scheme to reduce the discrepancy between source and target distributions was presented. Our approach is substantially different, as we do not restrict our attention to adapt a generic model to a specific scene and we tackle the problem of transferring knowledge among different modalities.

3.2.2 Learning Cross-modal Deep Representations

In the last few years deep networks have been successfully applied to learning feature representations from multi-modal data [63, 168, 165]. However, the problem of both learning and transferring cross-modal features has been rarely investigated. Notable exceptions are the works in [22, 137, 135, 46, 51]. Among these, the most similar to ours are [22, 137, 51]. In [22, 137] the idea of hallucinating data from other modalities was also exploited. However, our CNN-based approach is substantially different, since the work in [137] considered Deep Boltzmann Machines, while in [22] the mapping between different modalities was learned with Gaussian Processes. In [51] the problem of object detection from RGB data was addressed and depth images were used as additional information available only at training time. Similarly to [51], our detection network simultaneously use cross-modal features learned from a source domain and representations specific of the target scenario. However, in [51] labeled data were available in the original domain. Oppositely, in our framework we learn cross-modal features in an unsupervised setting, *i.e.* we do not require any annotation in the thermal domain. In this way, it is possible to exploit huge multispectral datasets.

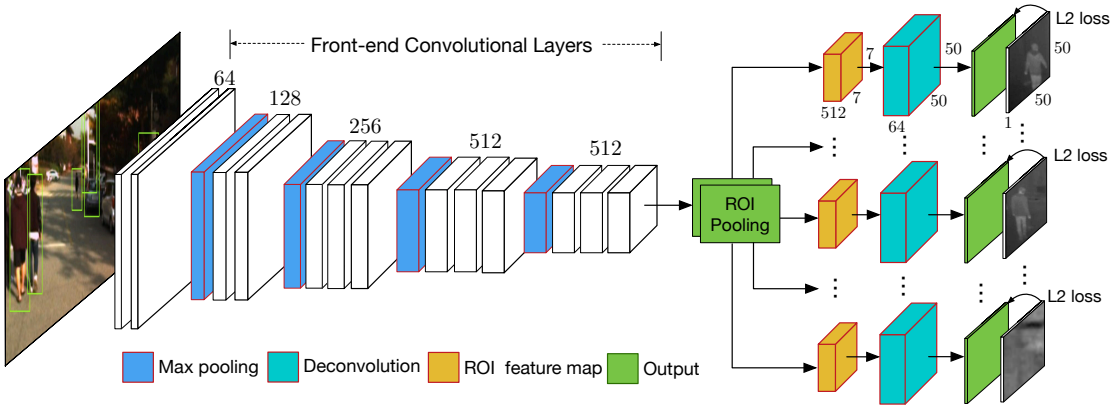


Figure 3.3: Architecture of the Region Reconstruction Network: a deep convolutional network trained for reconstructing thermal images from the associated RGB data. Best viewed in color.

3.3 Learning and transferring cross-modal deep representations

In this section we present the proposed framework. We first provide an overview of our approach and we describe in details the CNN architectures we design to reconstruct thermal data from RGB input and to transfer the learned cross-modal representations for the purpose of robust pedestrian detection.

3.3.1 Overview

As outlined in Section 3.1, the proposed framework (Fig.3.1) in based on two different convolutional neural networks, associated to the reconstruction and to the detection tasks, respectively. The first deep model, *i.e.* the Region Reconstruction Network (RRN), is a fully convolutional network trained on pedestrian proposals collected from RGB-thermal image pairs in an unsupervised manner. RRN is used to learn a non-linear mapping from the RGB channels to the thermal channel. In the target domain only RGB data are available and a second deep network, the Multi-Scale Detection Network (MSDN), embedding the parameters transferred from RRN, is used for robust pedestrian detection. MSDN takes a whole RGB image and a number of pedestrian proposals as input and outputs the detected bounding boxes with associated scores. In the test phase, detection is performed with MSDN and only RGB inputs are needed. In the following we describe the details of the proposed deepnet framework.

3.3.2 Region Reconstruction Network

The aim of RRN is to reconstruct thermal data from the associated RGB images. The design of the RRN architecture is driven by two main needs. First, in order to avoid human annotation efforts, thermal information should be recovered with an unsupervised approach. While our approach uses the thermal image as deep supervision for the reconstruction task, it essentially requires only very weak supervision information (*i.e.*, the pair-wise information). However, in the RGB-T data collection phase, we easily obtain the pair-wise information. The most expensive part in terms of human effort is to annotate the pedestrian bounding boxes. The proposed approach does not require these extra human-annotations. Second, as multispectral data are expected to be especially useful for hard positive and negative samples (Fig.3.2), instead of attempting to reconstruct the entire thermal images, it is more appropriate to specifically focus on bounding boxes which are likely to contain

pedestrians. Therefore, in this work we propose to exploit a pretrained generic pedestrian detector (*e.g.* ACF [27]) to extract a set of pedestrian proposals (containing true positives and false positives) from RGB data and design a deep model which reconstructs the associated thermal information.

The proposed RRN network is illustrated in Fig.3.3. The input of RRN is a three-channel RGB image and a set of associated pedestrian proposals. RRN consists of a front-end convolutional subnetwork and a back-end reconstruction subnetwork. Although in our implementation the front-end convolutional layers exploit the VGG-13 network structure [133], RRN alternatively supports other architectures. After the last convolutional layer of the front-end subnetwork, an ROI pooling layer [40] is added. For each ROI, feature maps with size $512 \times 7 \times 7$ are generated. Considering the small size of the ROI feature maps, in order to effectively reconstruct the regions of thermal images associated to pedestrians, we apply a deconvolutional layer to upsample the ROI feature maps (output size 50×50) and reduce the number of output channels to 64 to ensure smooth convergence during training. Different from many previous works (*e.g.* [158]) which simply consider a bilinear upsampling operator, in the deconvolutional layer we learn the upsampling kernels (kernel size 4, stride 8 and pad 1). After the deconvolutional layer, a Rectified Linear Unit (ReLU) layer is applied. Then, reconstruction maps corresponding to each proposal are generated using a convolutional layer (kernel size 3, pad 1). Finally, a square loss is considered to compute each reconstruction map and the whole network is optimized with back-propagation.

In the widely used Fast- or Faster-RCNN frameworks, the groundtruth pedestrian bounding boxes are used to determine the ratio of true positive and false positive samples, and then construct fixed-size training mini-batches. To avoid using the carefully annotated groundtruth bounding boxes, we construct each training mini-batch using pedestrian proposals generated by thresholded generic ACF from one randomly selected training image, since the number of the proposals corresponding to each training image dynamically changes, our approach thus implements a dynamic mini-batch size during training.

3.3.3 Multi-Scale Detection Network

MSDN is specifically designed to perform pedestrian detection from RGB images by exploiting the cross-modal representations learned with RRN. Inspired by previous works demonstrating the importance of considering multi-scale information in detection tasks [183, 161], we introduce a detection network which fuses multiple feature maps derived from ROI pooling layers.

MSDN architecture seamlessly integrates two sub-networks (Sub-Net A and Sub-Net B), as illustrated in Fig. 3.4. Sub-Net A has 13 convolutional layers, organized in five blocks. As depicted in Fig.3.4, $C_{m,n}$ denotes the m-th block with n convolutional layers with the same size filters. Max pooling layers are added after the convolutional layers, and the ReLU non-linearity is applied to the output of each convolutional layer. An RoI (Region of Interest) pooling layer [40] is applied to the last two convolutional blocks to extract feature maps of size $512 \times 7 \times 7$ for each pedestrian proposal. We consider these two blocks, as our experiments show that this strategy represents the optimal trade-off between computational complexity and accuracy. Sub-Net B has the same structure of Sub-Net A but, since its main goal is to transfer cross-modality mid-level representations, the parameters of the 13 convolutional layers ($C'_{1,2}$ to $C'_{5,3}$) are derived from the associated layers of RRN. Indeed, the convolutional blocks from RRN produce a compact feature

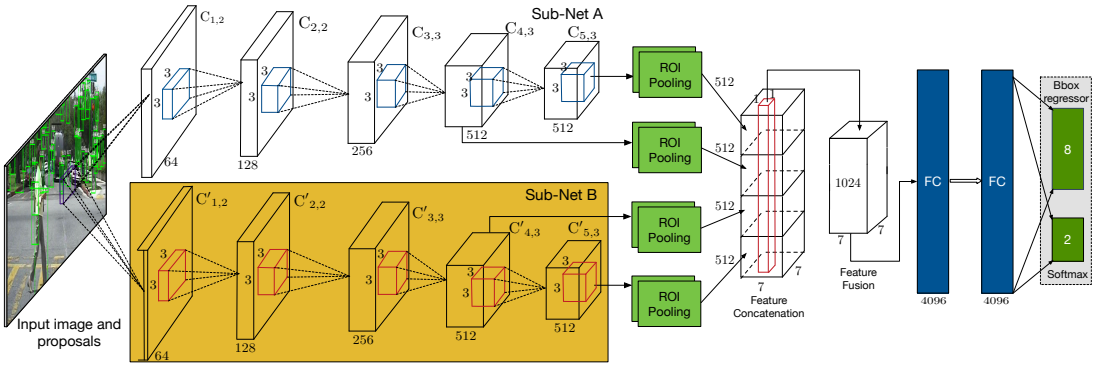


Figure 3.4: Architecture of the Multi-Scale Detection Network. Two sub-networks (Sub-Net A and Sub-Net B) with the same structure are used in MSDN. The parameters of all the convolutional layers of Sub-Net B (highlighted in yellow) are transferred from the Region Reconstruction Network.

representation which captures the complex relationship among the RGB and the thermal domain. Therefore, they are embedded in MSDN, such as to allow the desired knowledge transfer.

The feature maps derived from the RoI pooling layers of the two sub-networks are then combined with a concatenation layer and a further convolutional layer with 1024 channels is applied. As the size of the RoI feature maps is small, we set the kernel size equal to 1 in the convolutional layer. Then, two fully connected layers of size 4096 follow. Finally, two sibling layers are used, one that outputs softmax probability estimates over pedestrian and background classes, and another that provides the associated bounding-box offset values for pedestrian localization.

3.3.4 Optimization

As discussed above, the proposed cross-modal framework is based on two different deep networks. Therefore, the training process also involves two main phases.

In the first phase, RRN is trained on multispectral data. The front-end convolutional layers of RRN are initialized using the parameters of the 13 convolutional layers of the VGG-16 model [133] pretrained on ImageNet dataset. The remaining parameters are randomly initialized. Stochastic Gradient Descent (SGD) is used to learn the network parameters. In the second phase, the parameters of MSDN are optimized using RGB data and pedestrian bounding box annotations in the target domain. We first train Sub-Net A by adding the common parts of MSDN (*i.e.* from the feature concatenation layer to the two sibling layers). In this case the size of the feature maps in the concatenation and in the following convolutional layers is $1024 \times 7 \times 7$ and $512 \times 7 \times 7$, respectively. The pretrained VGG-16 model is also utilized to initialize Sub-Net A. The convolutional layers of Sub-Net B are initialized with the corresponding parameters of RRN. Then, fine-tuning is performed using the RGB data of the target domain. The whole MSDN optimization is based on back-propogation with SGD.

3.3.5 Pedestrian detection

In the detection phase, given a test RGB image, we adopt the standard protocol. First, region proposals are extracted, similarly to the training phase. Then, the input image and the proposals are fed into MSDN. The softmax layer outputs the class score and the bounding box regressor indicates the estimated image coordinates. To reduce the redundancy of the proposals, non-maximum suppression is employed based on the prediction

score of each proposal, setting an intersection over union (IoU) threshold δ.

3.4 Experiments

To evaluate the effectiveness of the proposed framework, we performed experiments on two publicly available datasets: the recent KAIST multispectral pedestrian dataset [57] and the popular Caltech pedestrian dataset [28]. In the following we describe the details of our evaluation.

3.4.1 Datasets

The **KAIST** multispectral pedestrian dataset [57] contains images captured under various traffic scenes with different illumination conditions (*i.e.* data recorded both during day and night). The dataset consists of 95,000 aligned RGB-thermal image pairs, of which 50,200 samples are used for training and the rest for testing. A total of 103,128 dense annotations corresponding to 1,182 unique pedestrians are available. We follow the protocol outlined in [57] in our experiments. The performance is evaluated on three different test sets, denoted as *Reasonable all*, *Reasonable day* and *Reasonable night*. *Reasonable* indicates that the pedestrians are not/partially occluded with more than 55 pixels height. The day and night sets are obtained from the *Reasonable all* set according to the capture time.

The **Caltech** pedestrian dataset [28] consists of about 10 hours of 30Hz video collected from a vehicle driving through urban traffic. The dataset contains 250,000 frames with 350,000 bounding boxes manually annotated and associated to about 2,300 unique pedestrians. Following previous works [145, 83], we strictly adopt the evaluation protocol in [28] measuring the log average miss rate over nine points ranging from 10^{-2} to 10^{0} False-Positive-Per-Image (FPPI). Our evaluation is conducted on both Caltech-All and Caltech-Reasonable settings.

Our approach uses RGB-thermal data for training, but in the test phase only requires RGB images as input. In all our experiments the KAIST training dataset is used to learn the RRN. Then, the performance of MSDN is assessed on the Caltech test set and on the RGB test frames of KAIST. The training and testing images of both datasets are resized (800 pixels height) to generate ROI feature maps with higher resolution useful for our reconstruction and detection tasks.

3.4.2 Experimental setup

Our framework is implemented under *Caffe*, and our evaluation is conducted on an Intel(R) Xeon(R) CPU E5-2630 with a single CPU core (2.40GHz), 64GB RAM and a NVIDIA Tesla K40 GPU.

We employ ACF [27] to generate pedestrian proposals for training both the reconstruction and the detection network with a low detection threshold of -70 as in [83] to obtain a high recall of pedestrian regions. In the test phase we also use ACF and consider the test proposals available online[2]. It is worth nothing that, while we focus on ACF, our cross-modality learning approach can be used in combination with an arbitrary proposal method.

For training the reconstruction network, we use the whole training set of the KAIST dataset. As thermal images captured from an infrared device have relatively low contrast and significant noise, we perform some basic processing, such as adaptive histogram equalization and denoising. By computing pedestrian proposals applying ACF, we end up

[2] `http://www.vision.caltech.edu/Image$_$Datasets/CaltechPedestrians/`

Figure 3.5: KAIST dataset. Reconstructed regions of thermal images (50×50 pixels) associated to the top nine detected pedestrian windows from ACF.

Methods	All	Day	Night
CMT-CNN-SA	54.26%	52.44%	58.97%
CMT-CNN-SA-SB(Random)	56.76%	54.83%	61.24%
CMT-CNN-SA-SB(ImageNet)	52.15%	50.71%	57.65%
CMT-CNN	49.55%	47.30%	54.78%

Table 3.1: Comparison of different methods on the KAIST multispectral datasets including reasonable all, reasonable day and reasonable night settings.

creating a dataset of about 20K frames for training the region reconstruction network. All the frames are then horizontally flipped for data augmentation. We generate mini-batch of reconstruction RoIs from randomly chosen two images, and a fixed learning rate $\lambda_r = 10^{-9}$ is used to guarantee smooth convergence. We train the RRN for about 10 epochs.

For training the detection network on the the Caltech dataset we follow [185] and we construct a training set where every 3^{rd} frame is used. Instead, for the KAIST dataset we adopt the standard training protocol and every 20^{th} frame is considered. For both datasets, we use the same protocol for training MSDN. Similarly to RRN training, the data are flipped horizontally for the purpose of data augmentation. Each mini-batch consists of 128 pedestrian proposals randomly chosen from one training image. Positive samples with a ratio of 25% are taken from the proposals which have an IoU overlap with the ground truth of more than 0.5, while negative samples are obtained when the IoU overlap is in the range of [0, 0.5]. Stochastic gradient descent is used to optimize MSDN with the momentum and the weight decay parameters set to 0.9 and 0.0005, respectively. The network is trained for 8 epochs using an initial learning rate of 0.001 and drop by 10 times at the 5^{th} epoch.

3.4.3 Results on KAIST multispectral dataset

Analysis of proposed method. The first series of experiments aims to demonstrate the effectiveness of the proposed Cross-Modality Transfer CNN (CMT-CNN) framework. We

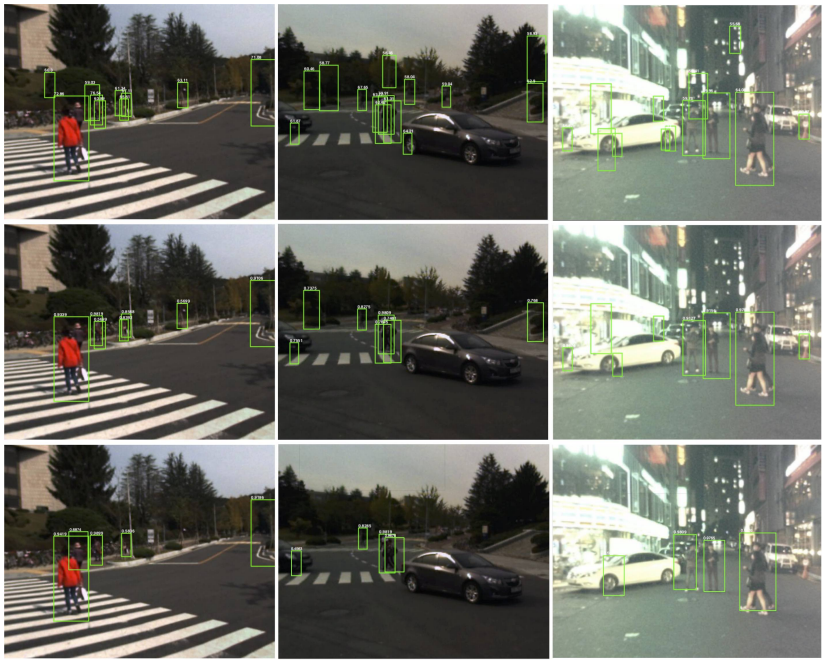
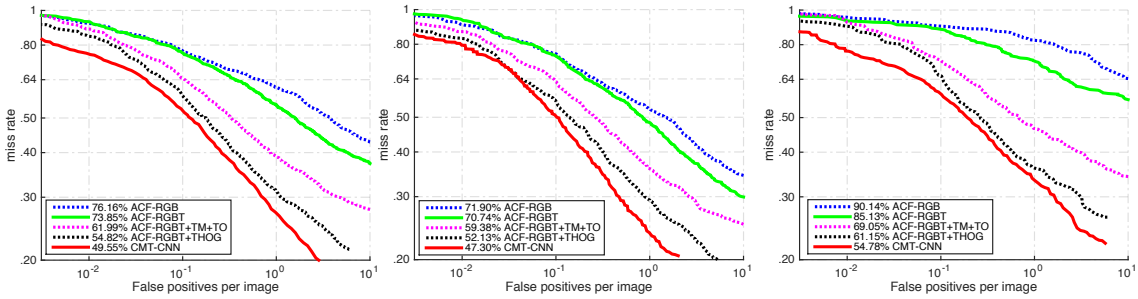


Figure 3.6: Examples of pedestrian detection results under different illumination conditions on the KAIST multispectral pedestrian dataset: (top) ACF detector, (middle) CMT-CNN-SA, (bottom) CMT-CNN.



(a) Reasonable all (b) Reasonable day (c) Reasonable night

Figure 3.7: Quantitative evaluation results (miss rate versus false positive per image) on the KAIST multispectral dataset.

evaluate the performance of our approach under four different settings: (i) CMT-CNN-SA. We only use Sub-Net A. The two ROI feature maps are concatenated and given as input to the convolutional fusion layer. This layer outputs a feature map with size 512, rather than 1024. Finally, the output is fed to the fully connected layers; (ii) CMT-CNN-SA-SB (ImageNet). We consider two sub-networks but initialize the convolutional layers of Sub-Net B from pretrained VGG16 model on ImageNet; (iii) CMT-CNN-SA-SB (Random): Same as (ii) but with random initialization for Sub-Net B; (v) CMT-CNN as described in Section 3.3, *i.e.* initializing the convolutional layers of Sub-Net B from trained RRN.

Table 3.1 shows the results of our comparison. Performance is evaluated using the log average miss-rate (MR). From the table it is clear that CMT-CNN significantly outperforms all its variations on all the three test sets, confirming the fact that the proposed cross-modality framework improves the detection accuracy. We also observe that CMT-CNN provides lower MR than CMT-CNN-SA-SB, indicating that the performance gain of

Methods	Miss-Rate
CMT-CNN-SA	13.76%
CMT-CNN-SA-SB(Random)	15.89%
CMT-CNN-SA-SB(ImageNet)	13.01%
CMT-CNN-SA-SB(RGB-KAIST)	12.51%
CMT-CNN	10.69%

Table 3.2: Comparison of different variants of our method on the Caltech-Reasonable dataset. Performance are evaluated in terms of log-average miss-rate.

batch size	32	64	128	256
Caltech-All	65.97%	65.68%	65.32%	65.42%
Caltech-Reasonable	13.52%	13.01%	12.51%	12.35%

Table 3.3: Performance using different batch size in CMT-CNN-SA-SB (RGB-KAIST) experiments.

CMT-CNN is not only due to an increased number of parameters.

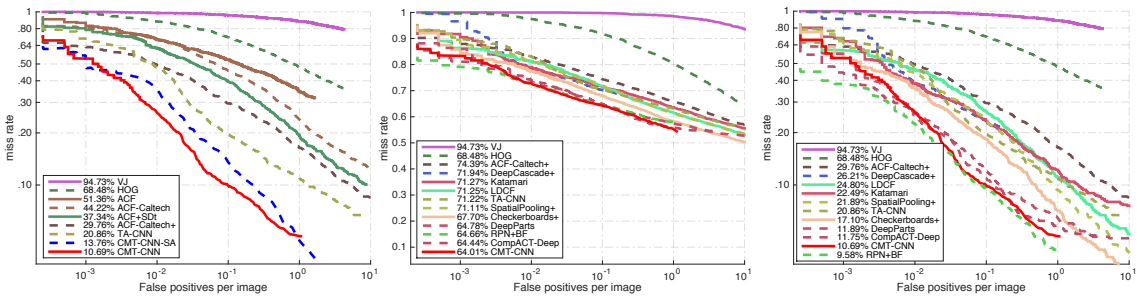
Figure 3.5 depicts some examples of the reconstruction results obtained with the proposed RRN. For the two given test frames, the reconstructed thermal regions associated to the top nine detection windows computed with ACF are shown. From the figure, it is easy to observe that the proposed network is able to effectively learn a mapping from RGB data to thermal data. Figure 3.6 shows some qualitative results obtained with MSDN. Comparing the detection bounding boxes of CMT-CNN-SA with those of CMT-CNN, we observe that hard negative samples are correctly classified with our method. For instance, the foliage from the trees (Fig. 3.6- first and second columns) is wrongly detected as pedestrian by CMT-CNN-SA. This confirms our intuition that leveraging information from multispectral data with our cross-modal representation transfer approach permits to improve the detection accuracy.

Comparison with state of the art methods. We also compare our approach with state of the art methods on the KAIST multispectral dataset. These methods include: (i) ACF-RGB [27], *i.e.* using ACF on RGB data; (ii) ACF-RGBT [57], *i.e.* using ACF on RGB-Thermal data; (iii) ACF-RGBT+TM+TO [57], *i.e.* using ACF on RGB-Thermal data with extra gradient magnitude and HOG of thermal images; (iv) ACF-RGBT+HOG [57], *i.e.* using ACF on RGB-Thermal data with HOG features with more gradient orientations than (iii). Results associated to these methods have been taken directly from the original paper [57]. Similarly to baseline approaches, we also use ACF for generating proposals both at training and at test time.

Observing Fig. 3.7, it is clear that CMT-CNN is several points better than ACF-RGBT+HOG, the best baseline on the KAIST dataset. Importantly, CMT-CNN only uses color images in the test phase, while ACF-RGBT+HOG exploits both RGB and thermal data. We also observe that on the *Reasonable night* setting, our approach obtains a more significant improvement than in the *Reasonable day* experiments. This demonstrates that CMT-CNN is especially useful for pedestrian detection under dark illumination conditions, thus confirming our initial intuition.

3.4.4 Results on Caltech pedestrian dataset

Analysis of CMT-CNN. Similarly to the experiments on the KAIST dataset, we first analyze the performance of our approach when different initialization strategies are used for Sub-Net B. In this case we also consider another baseline CMT-CNN-SA-SB (RGB-KAIST),



(a) (b) (c)

Figure 3.8: Quantitative evaluation results on the Caltech pedestrian dataset: comparison with (a) previous methods using ACF for proposals (VJ and HOG methods do not use ACF, but are kept as reference points) (b) state of the art methods on Caltech-All (c) state of the art methods on Caltech-Reasonable.

Method	Hardware	Miss-Rate	Testing Time (s/f)
InformedHaar [184]	CPU	75.85%	1.59
SpatialPooling [108]	CPU	74.04%	7.69
LDCF [102]	CPU	71.25%	0.60
CCF [173]	Titan Z GPU	66.73%	13.0
RPN + BF [183]	Tesla K40 GPU	64.66%	0.51
CompACT-Deep [14]	Tesla K40 GPU	64.44%	0.50
CMT-CNN	Tesla K40 GPU	64.01%	0.59

Table 3.4: Comparison of different methods (log-average miss-rate vs detection time). Log-average miss-rate is evaluated on the Caltech-All. s/f represents seconds per frame.

i.e. we initialize Sub-Net B with VGG16 pretrained on ImageNet and further train it using RGB data of KAIST. The results of the comparison are shown in Table 3.2 and confirm the effectiveness of our framework. We observe that CMT-CNN-SA-SB (RGB-KAIST) beats CMT-CNN-SA-SB (ImageNet), showing that fine tuning CMT-CNN-SB with KAIST RGB data provides effective representations for improving the detection performance on Caltech. By using complementary data from the thermal modality, CMT-CNN further boosts its accuracy and outperforms CMT-CNN-SA-SB (RGB-KAIST). We observe that the improvement due to knowledge transfer on Caltech data is less pronounced than that obtained on KAIST dataset. We believe that this is mainly due to the fact that the frames of Caltech generally exhibit better illumination conditions than those of KAIST, while thermal information is especially beneficial in case of bad illumination.

To further demonstrate that the performance gain obtained with the proposed CMT-CNN is not simply due to ensembling different models, we consider the baseline CMT-CNN-SA-SB(RGB-KAIST) and we train Sub-Net B with KAIST RGB images using four different mini-batch size ranging from 32 to 256. For each experiment, the training samples are randomly shuffled. Table 3.3 shows the results of the four trials on Caltech-All and Caltech-Reasonable: using different batch size for Sub-Net B slightly affects the final performance and the best MR reported in the table is still worse than those obtained with CMT-CNN. This confirms the validity of our cross-modality learning approach.

We also compare the proposed CMT-CNN which uses ACF to generate region proposals with previous approaches also based on ACF proposals. Figure 3.8(a) shows the results of our comparison: our model outperforms all the baselines. Moreover, similarly to what

we observed for KAIST experiments, CMT-CNN is more accurate than CMT-CNN-SA, confirming the advantage of our approach.

Comparison with state of the art methods. A comparison with state of the art methods is provided in Fig. 3.8(b). We considered Viola-Jones (VJ) [147], Histograms of Oriented Gradients (HOG) [24], DeepCascade+ [3], LDCF [102], SCF+AlexNet [54], Katamari [8], SpatialPooling+ [109], SCCPriors [175], TA-CNN [145], CCF and CCF+CF [173], Checkerboards and Checkerboards+ [185], DeepParts [144], CompACT-Deep [14] and RPN+BF[183]. Our approach attains a miss-rate of 10.69% on Caltech-Reasonable, which is very competitive with the state of the art methods, and a miss-rate of 64.01% on Caltech-All, which establishes a new state-of-the-art result. Importantly, our approach can be seen as complementary to most previous works. In fact, we believe that our unsupervised learning of cross-modal representations can be also integrated in other CNN architectures, to improve their robustness in coping with bad illumination conditions.

In Table 3.4 we report a comparison between our framework and recent pedestrian detection methods in terms of computational efficiency (times associated to previous methods are taken from the original papers). At test time, our network takes only 0.59 seconds to process one image, which is very competitive with previous methods.

3.5 Conclusion

We presented a novel approach for robust pedestrian detection under adverse illumination conditions. Inspired by previous works on multi-scale pedestrian detection [183], a novel deep model is introduced to learn discriminative feature representations from raw RGB images. Differently form previous methods, the proposed architecture integrates a sub-network, pre-trained on pairs of RGB and thermal images, such as to learn cross-modal feature representations. In this way, knowledge transfer from multispectral data is achieved and accurate detection is possible even in case of challenging illumination conditions. The effectiveness of the proposed approach is demonstrated with extensive experiments on publicly available benchmarks: the KAIST multispectral and the Caltech pedestrian detection datasets.

While this work specifically addresses the problem of pedestrian detection, the idea behind our cross-modality learning framework can be useful in other applications (*e.g.* , considering reconstructing depth images [164] for RGBD object/action detection and recognition). Hence, natural directions for future research include further investigating this possibility.

CHAPTER 4

Multi-Scale Structured Prediction and Fusion via Continuous CRFs as Sequential Deep Networks for Monocular Depth Estimation [1]

Depth cues have been proved very useful in various computer vision and robotic tasks. This chapter addresses the problem of monocular depth estimation from a single still image. Inspired by the effectiveness of recent works on multi-scale convolutional neural networks (CNN), we propose a deep model which fuses complementary information derived from multiple CNN side outputs. Different from previous methods using concatenation or weighted average schemes, the integration is obtained by means of continuous Conditional Random Fields (CRFs). In particular, we propose two different variations, one based on a cascade of multiple CRFs, the other on a unified graphical model. By designing a novel CNN implementation of mean-field updates for continuous CRFs, we show that both proposed models can be regarded as sequential deep networks and that training can be performed end-to-end. Through an extensive experimental evaluation, we demonstrate the effectiveness of the proposed approach and establish new state of the art results for the monocular depth estimation task on three publicly available datasets, *i.e.* NYUD-V2, Make3D and KITTI.

4.1 Introduction

While estimating the depth of a scene from a single image is a natural ability for humans, devising computational models for accurately predicting depth information from RGB data is a challenging task. Many attempts have been made to address this problem in the past. In particular, recent works have achieved remarkable performance thanks to powerful deep learning models [29, 30, 89, 112]. Assuming the availability of a large training set of RGB-depth pairs, monocular depth prediction from single images can be regarded as

[1] Dan Xu, Elisa Ricci, Wanli Ouyang, Xiaogang Wang, Nicu Sebe, "Multi-Scale Continuous CRFs as Sequential Deep Networks for Monocular Depth Estimation", IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017). Dan Xu, Elisa Ricci, Wanli Ouyang, Xiaogang Wang, Nicu Sebe, "Monocular Depth Estimation using Multi-Scale Continuous CRFs as Sequential Deep Networks", IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI) (in press), 2018.

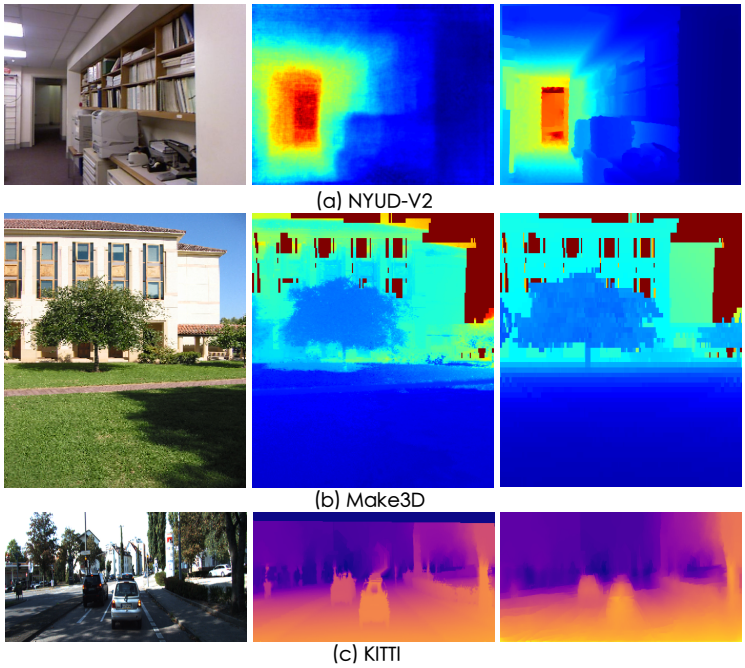


Figure 4.1: Monocular depth estimation results on three different benchmark datasets, *i.e.* NYUD-V2 (the 1st row), Make3D (the 2nd row) and Kitti (the 3rd row), using the proposed multi-scale CRF model with a pre-trained CNN (*e.g.* VGG Convolution-Deconvolution [104]). From left to right, each column is original RGB images, the recovered depth maps and the groundtruth, respectively.

a pixel-level continuous regression problem and Convolutional Neural Network (CNN) architectures are typically employed.

In the last few years significant efforts have been made in the research community to improve the performance of CNN models for pixel-level prediction tasks (*e.g.* semantic segmentation, contour detection). Previous works have shown that, for depth estimation as well as for other pixel-level classification or regression problems, more accurate estimates can be obtained by combining information from multiple scales [29, 158, 20]. This can be achieved in different ways, *e.g.* fusing feature maps corresponding to different network layers or designing an architecture with multiple inputs corresponding to images at different resolutions. Other works have demonstrated that, by adding a Conditional Random Field (CRF) in cascade to a convolutional neural architecture, the performance can be greatly enhanced and the CRF can be fully integrated within the deep model enabling end-to-end training with back-propagation [188]. However, these works mainly focus on pixel-level prediction problems in the discrete domain (*e.g.* semantic segmentation). While complementary, so far these strategies have been only considered in isolation and no previous works have exploited multi-scale information within a CRF inference framework.

In this work we argue that, benefiting from the flexibility and the representational power of graphical models, we can optimally fuse representations derived from multiple CNN side-output layers using structured constraints, improving performance over traditional multi-scale strategies. By exploiting this idea, we introduce a novel framework to estimate depth maps from single still images. Opposite to previous work fusing multi-scale features by weighted averaging or concatenation, we propose to integrate multi-layer side-output information by designing a novel approach based on continuous CRFs. Specifically, we present two different methods. The first approach is based on a single multi-scale unified

CRF model, while the other considers a cascade of scale-specific CRFs. We also show that, by introducing a common CNN implementation for mean-fields updates in continuous CRFs, both models are equivalent to sequential deep networks and an end-to-end approach can be devised for training. Through extensive experimental evaluation we demonstrate that the proposed CRF-based approach produces more accurate depth maps than traditional multi-scale approaches for pixel-level prediction tasks [49, 158]. Moreover, by performing experiments on the publicly available NYU Depth V2 [132], Make3D [127] and KITTI [38] datasets, we show that our approach is able to robustly reconstruct depth with good visual quality (Fig.5.1) and outperforms state of the art methods for the monocular depth estimation task.

This paper extends our earlier work [164] through proposing and investigating different multi-scale connection structures for message passing, further enriching the related works, providing more approach details, and significantly expanding experimental results and analysis. To summarize, the contribution of this paper is threefold:

- Firstly, we propose a novel approach for predicting depth maps from RGB inputs which exploits multi-scale estimations derived from CNN inner semantic layers by structurally fusing them within a unified CNN-CRF framework.

- Secondly, as the task of pixel-level depth prediction implies inferring a set of continuous values, we show how mean field (MF) updates can be implemented as sequential deep models, enabling end-to-end training of the whole network. We believe that our MF implementation will be useful not only to researchers working on depth prediction, but also to those interested in other problems involving continuous variables. Therefore, our code is made publicly available at https://github.com/danxuhk/ContinuousCRF-CNN.git.

- Thirdly, our experiments demonstrate that the proposed multi-scale CRF framework is superior to previous methods integrating information from different semantic network layers by combining multiple losses [158] or by adopting feature concatenations [49]. We also show that our approach outperforms state of the state of the art monocular depth estimation methods on public benchmarks and that the proposed CRF-based models can be employed in combination with different pre-trained CNN architectures, consistently enhancing their performance.

The remainder of this paper is organised as follows. We first introduce related work in Section 4.2, and then the proposed multi-scale CRF models for monocular depth estimation is presented in Section 4.3. We further elaborate how the proposed models can be implemented as sequential neural network for end-to-end joint optimization in Section 4.4. The experimental results and analysis are elaborated in Section 4.5, and we conclude the paper in Section 4.6.

4.2 Related work

Our approach is built upon recent successes of deep CNN architectures for image classification [71, 133, 50] and fully convolutional networks for dense semantic image segmentation [94, 104]. We briefly introduce the most related works by organizing them into three main aspects, *i.e.* monocular depth estimation, multi-scale CNN and dense pixel-level prediction via combination of CNN and CRFs.

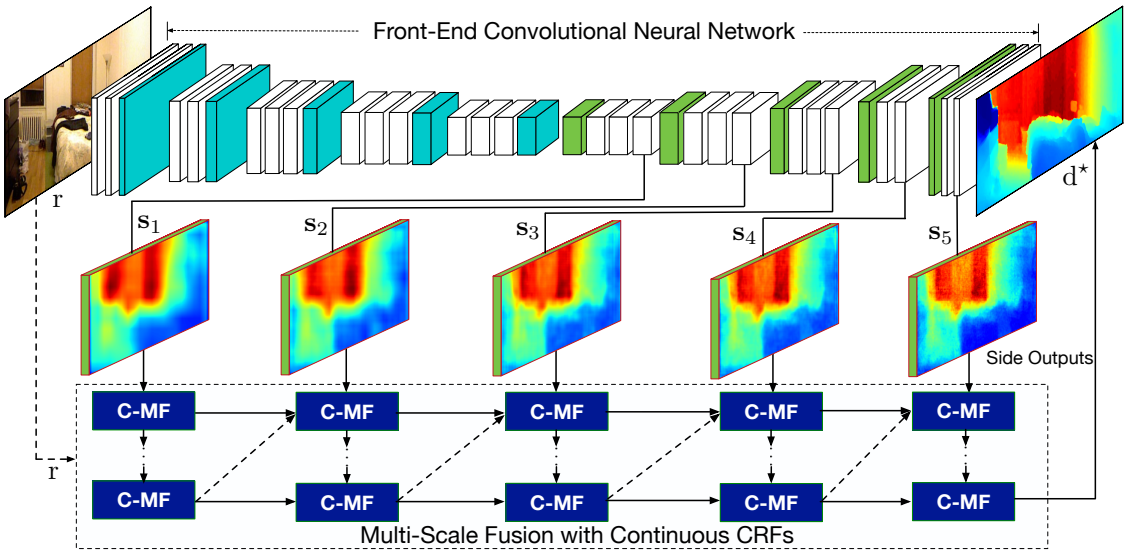


Figure 4.2: Overview of the proposed deep architecture. Our model is composed of two main components: a front-end CNN and a fusion module. The fusion module uses continuous CRFs to integrate multiple side output maps of the front-end CNN. We consider two different CRFs-based multi-scale models and implement them as sequential deep networks by stacking several elementary blocks, the C-MF blocks.

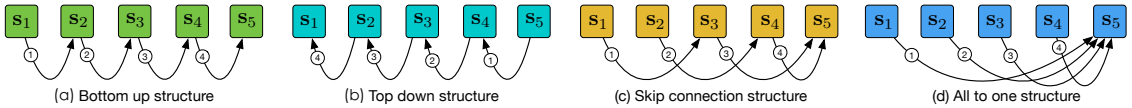


Figure 4.3: Illustration of different multi-scale message passing structures for the integration of the multi-scale predictions $\mathbf{s}_1$ to $\mathbf{s}_5$ produced from the front-end convolutional network. The arrows represent the direction of the message passing, and the numbers in circles represent the order. The dashed line box in Fig. 5.2 shows a bottom-up passing structure.

4.2.1 Monocular Depth Estimation.

Previous approaches for depth estimation from single images can be grouped into three main categories: (i) methods operating on hand crafted features, (ii) methods based on graphical models and (iii) methods adopting deep convolutional neural networks.

Earlier works addressing the depth prediction task belong to the first category. Hoiem *et al.*[52, 53] proposed photo pop-up, a fully automatic method for creating a basic 3D model from a single photograph by introducing an assumption of 'ground-vertical' geometric structure. Karsch *et al.*[64] developed Depth Transfer, a non parametric approach based on SIFT Flow, where the depth of an input image is reconstructed by transferring the depth of multiple similar images and then applying some warping and optimizing procedures. Instead of directly recovering depth from appearance features, Liu *et al.* [88] explored using semantic scene segmentation results to guide the 3-D depth reconstruction. Similarly, Ladicky *et al.*[75] also demonstrated the benefit of combining semantic object labels with depth features. However, the hand-crafted representations are not robust enough for this challenging problem.

In the second category, some works exploited the flexibility of graphical models to reconstruct depth information. For instance, Delage *et al.*[25] proposed a dynamic Bayesian framework for recovering 3D information from indoor scenes. A discriminatively-trained multiscale Markov Random Fields (MRFs) were introduced in [126, 125], in order to

optimally fuse local and global features. Depth estimation was treated as an inference problem in a discrete-continuous CRF model in [92]. However, these works did not employ deep networks.

More recent approaches for depth estimation are based on CNNs [29, 89, 153, 119, 77]. For instance, Eigen *et al.*[30] proposed a multi-scale approach for depth prediction, considering two deep networks, one performing a coarse global prediction based on the entire image, and the other refining predictions locally. This approach was extended in [29] to handle multiple tasks (*e.g.* semantic segmentation, surface normal estimation). Wang *et al.*[153] introduced a CNN for joint depth estimation and semantic segmentation. The obtained estimates were further refined with Hierarchical CRFs. The most similar work to ours is [89], where the representational power of deep CNN and continuous CRFs is jointly exploited for depth prediction. However, the method proposed in [89] is based on superpixels and the information associated to multiple scales is not exploited in their graphical model.

4.2.2 Multi-scale CNNs.

The problem of combining information from multiple scales has recently received considerable interest in various computer vision tasks. In [158] a deeply supervised fully convolutional neural network was proposed for edge detection by weighted combination of multiple side outputs. Skip-layer networks, where the feature maps derived from different semantic layers of a primary front-end network are jointly considered in an output layer, have also become very popular [94, 10]. Other works considered multi-stream architectures, where multiple parallel networks receiving inputs at different scale are fused [12]. Cai *et al.* [13] proposed a multi-scale method via combining the predictions obtained from feature maps with different resolution for object detection. Dilated convolutions (*e.g. dilation* or *à trous*) have been also employed in different deep network models in order to aggregate multi-scale contextual information [18]. However, in these works, the multi-scale representations or estimations are typically combined by using simple concatenation or weighted averaging operation. We are not aware of previous works exploring fusing deep multi-scale information within a CRF framework.

4.2.3 Dense Pixel-level Prediction via Combination of CNN and CRFs.

The combination of CNN and CRFs has shown great usefulness for dense pixel-level structured prediction [128, 67]. Some existing works utilize CRFs as a post processing module for further refining the predictions from the CNN [19, 110]. To benefit from end-to-end learning, Zhang *et al.* [188] proposed a CRF-RNN model which jointly optimizes a front-end deep network with a discrete CRF for semantic image segmentation. However, as far as we know, this work is a first attempt to combine multi-scale continuous CRFs with deep convolutional neural network for constructing a unified model for end-to-end monocular depth estimation.

4.3 Multi-Scale CRF Models for Monocular Depth Estimation

In this section we introduce our deep model with the designed multi-scale continuous CRFs for monocular depth estimation from single images. We first formalize the problem of depth prediction and give a brief overview of the proposed approach. Then, we describe two different variants of the proposed multi-scale model, one based on a cascade of CRFs

and the other on a single multi-scale unified CRFs.

4.3.1 Problem Formulation and Overview

Following previous works we formulate the task of depth prediction from monocular RGB input as the problem of learning a non-linear mapping $F : \mathcal{I} \rightarrow \mathcal{D}$ from the image space $\mathcal{I}$ to the output depth space $\mathcal{D}$. More formally, let $\mathcal{Q} = \{(\mathbf{r}_i, \bar{\mathbf{d}}_i)\}_{i=1}^{Q}$ be a training set of Q pairs, where $\mathbf{r}_i \in \mathcal{I}$ denotes an input RGB image with N pixels and $\bar{\mathbf{d}}_i \in \mathcal{D}$ represents its corresponding real-valued depth map.

For learning F we consider a deep model made of two main building blocks (Fig. 5.2). The first component is a CNN architecture with a set of intermediate side outputs $\mathcal{S} = \{\mathbf{s}_l\}_{l=1}^{L}$, $\mathbf{s}_l \in R^N$, produced from L different layers with a mapping function $f_s(\mathbf{r}; \boldsymbol{\Theta}, \boldsymbol{\theta}_l) \rightarrow \mathbf{s}_l$. For simplicity, we denote with $\boldsymbol{\Theta}$ the set of front-end network layer parameters and with $\boldsymbol{\theta}_l$ the parameters of the network branch producing the side output associated to the l-th layer (see Section 4.5.1 for details of our implementation). In the following we denote this network as the front-end CNN.

The second component of our model is a fusion block. As shown in previous works [94, 10, 158], features generated from different CNN layers capture complementary information. The main idea behind the proposed fusion block is to use CRFs to effectively integrate the side output maps of our front-end CNN for robust depth prediction. Our approach develops from the intuition that these representations can be combined within a sequential framework, *i.e.* performing depth estimation at a certain scale and then refining the obtained estimates in the subsequent level. Specifically, we introduce and compare two different multi-scale models, both based on CRFs, and corresponding to two different versions of the fusion block. The first model is based on a **single multi-scale unified CRFs**, which integrates information available from different scales and simultaneously enforces smoothness constraints between the estimated depth values of neighboring pixels and neighboring scales. The second model implements a **cascade of scale-specific CRFs**: at each scale l a CRF is employed to recover the depth information from side output maps $\mathbf{s}_l$ and the outputs of each CRF model are used as additional observations for the subsequent model. In Section 4.3.2 we describe the two models in details, while in Section 4.4 we show how they can be implemented as sequential deep networks by stacking several elementary blocks. We call these blocks C-MF blocks, as they implement Mean Field updates for Continuous CRFs (Fig. 5.2).

4.3.2 Multi-scale Fusion with Continuous CRFs

We now elaborate the proposed CRF-based models for fusing multi-scale side-outputs derived from different semantic layers of the front-end deep convolutional neural networks.

Multi-Scale Unified CRF Model

Given a vector $\hat{\mathbf{s}}$ with a dimension of $L \times N$ obtained by concatenating the side output score maps $\{\mathbf{s}_1, \ldots, \mathbf{s}_L\}$ and a vector $\mathbf{d}$ with a dimension of $L \times N$ expressing real-valued output variables, we define a CRF modeling the following conditional distribution:

$$P(\mathbf{d}|\hat{\mathbf{s}}) = \frac{1}{Z(\hat{\mathbf{s}})} \exp\{-E(\mathbf{d}, \hat{\mathbf{s}})\}, \tag{4.1}$$

where $Z(\hat{\mathbf{s}}) = \int_{\mathbf{d}} \exp\{-E(\mathbf{d}, \hat{\mathbf{s}})\} d\mathbf{d}$ is the partition function [76] acting as a normalization factor for probabilities. The energy function is defined as:

$$E(\mathbf{d}, \hat{\mathbf{s}}) = \sum_{i=1}^{N} \sum_{l=1}^{L} \phi(d_i^l, \hat{\mathbf{s}}) + \sum_{i,j} \sum_{l,k} \psi(d_i^l, d_j^k), \tag{4.2}$$

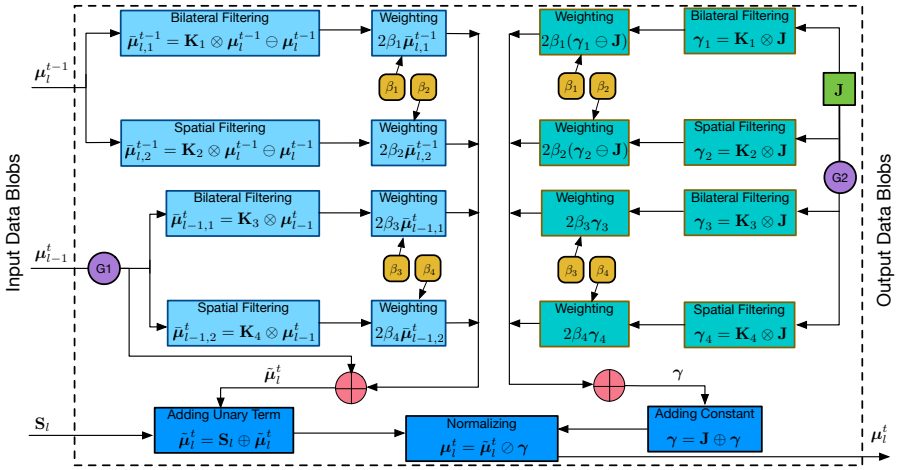


Figure 4.4: Detailed computing flow graph of the proposed C-MF block. $\mathbf{J}$ represents a $W \times H$ matrix with all elements equal to one. The symbols $\oplus$, $\ominus$, $\oslash$ and $\otimes$ indicate element-wise addition, subtraction, division and Gaussian convolution operation, respectively. G1 and G2 represent two gate functions for controlling the computing flow.

and d_i^l indicates the hidden variable associated to scale l and pixel i. The first term is the sum of quadratic unary terms defined as:

$$\phi(d_i^l, \hat{\mathbf{s}}) = (d_i^l - s_i^l)^2, \tag{4.3}$$

where s_i^l is the regressed depth value at pixel i and scale l obtained with $f_s(\mathbf{r}; \boldsymbol{\Theta}, \boldsymbol{\theta}_l)$. The second term is the sum of pairwise potentials describing the relationship between pairs of hidden variables d_i^l and d_j^k and is defined as follows:

$$\psi(d_i^l, d_j^k) = \sum_{m=1}^{M} \beta_m w_m(i,j,l,k,\mathbf{r})(d_i^l - d_j^k)^2, \tag{4.4}$$

where $w_m(i,j,l,k,\mathbf{r})$ is a weight which specifies the relationship between the estimated depth of the pixels i and j at scale l and k, respectively; M is the number of kernels.

To perform inference we rely on the mean-field theory to approximate $P(\mathbf{d}|\hat{\mathbf{s}})$ with another distribution $Q(\mathbf{d}|\hat{\mathbf{s}})$, where $Q(\mathbf{d}|\hat{\mathbf{s}}) = \prod_{i=1}^{N} \prod_{l=1}^{L} Q_{i,l}(d_i^l|\hat{\mathbf{s}})$, expressing a product of independent marginals. By minimizing the Kullback-Leibler divergence between the distribution of P and Q, we obtain the solution of Q. As the log distribution $\log Q_{i,l}(d_i^l|\hat{\mathbf{s}})$ has a quadratic form w.r.t. d_i^l and can be represented as Gaussian distribution, the following mean-field updates can be derived:

$$\gamma_{i,l} = 2(1 + 2\sum_{m=1}^{M} \beta_m \sum_{k} \sum_{j,i} w_m(i,j,l,k,\mathbf{r})), \tag{4.5}$$

$$\mu_{i,l} = \frac{2}{\gamma_{i,l}}(s_i^l + 2\sum_{m=1}^{M} \beta_m \sum_{k} \sum_{j,i} w_m(i,j,l,k,\mathbf{r})\mu_{j,k}). \tag{4.6}$$

Here $\gamma_{i,l}$ and $\mu_{i,l}$ are the variance and mean of the distribution $Q_{i,l}$, respectively.

To define the weights $w_m(i,j,l,k,\mathbf{r})$ we introduce the following assumptions. First, we assume that the estimated depth at scale l only depends on the depth estimated at previous scale. Second, for relating pixels at the same and at previous scale, we set weights depending on m kernel functions K_m^{ij}, which consists of Gaussian kernels with form of $\exp\big(-\frac{\|\mathbf{h}_i^m - \mathbf{h}_j^m\|_2^2}{2\theta_m^2}\big)$. Here, $\mathbf{h}_i^m$ and $\mathbf{h}_j^m$ indicate some features derived from the input image $\mathbf{r}$ for pixels i and j. θ_m are user-defined bandwidth parameters [68]. Following previous

works [188, 68], we use pixel positions and color values as features, leading to two kernel functions, *i.e.* a bilateral appearance kernel using both the pixel positions and the color value features and a spatial smoothness kernel using only the pixel positions features, for modeling dependencies of pixels at scale l and other two for relating pixels at neighboring scales. Under these assumptions, the mean-field updates (4.5) and (4.6) can be rewritten as:

$$\gamma_{i,l} = 2\big(1 + 2\sum_{m=1}^{2}\beta_m \sum_{j\neq i} K_m^{ij} + 2\sum_{m=3}^{4}\beta_m \sum_{j,i} K_m^{ij}\big), \tag{4.7}$$

$$\begin{aligned}\mu_{i,l} = \quad \frac{2}{\gamma_{i,l}}\big(s_i^l + 2\sum_{m=1}^{2}\beta_m \sum_{j\neq i} K_m^{ij}\mu_{j,l}, \\ +2\sum_{m=3}^{4}\beta_m \sum_{j,i} K_m^{ij}\mu_{j,l-1}\big).\end{aligned} \tag{4.8}$$

The parameters β_m need to be learned during training. We will present the details of the parameter optimization in Section 4.4. Given a new test image, the optimal $\tilde{\mathbf{d}}$ can be computed via maximizing the log conditional probability [118], *i.e.* $\tilde{\mathbf{d}} = \arg\max_{\mathbf{d}} \log(Q(\mathbf{d}|\mathbf{S}))$, where $\tilde{\mathbf{d}} = [\mu_{1,1}, ..., \mu_{N,L}]$ is a vector of the $L \times N$ mean values associated to $Q(\mathbf{d}|\hat{\mathbf{s}})$. We take the estimated variables at the finest scale L (*i.e.* $\mu_{1,L}, ..., \mu_{N,L}$) as our predicted depth map $\mathbf{d}^\star$.

Multi-Scale Cascade CRF Model

The cascade model is based on a set of L CRF models, each one associated to a specific scale l, which are progressively stacked such that the estimated depth at previous scale can be used as observations of the CRF model in the following scale level. Each CRF is used to compute the output vector $\mathbf{d}^l$ and it is constructed considering the side output representations $\mathbf{s}^l$ and the estimated depth at the previous step $\tilde{\mathbf{d}}^{l-1}$ as observed variables, *i.e.* $\mathbf{o}^l = [\mathbf{s}^l, \tilde{\mathbf{d}}^{l-1}]$. The associated energy function of the CRF model is defined as:

$$E(\mathbf{d}^l, \mathbf{o}^l) = \sum_{i=1}^{N}\phi(d_i^l, \mathbf{o}^l) + \sum_{i\neq j}\psi(d_i^l, d_j^l). \tag{4.9}$$

The unary and pairwise terms can be defined analogously to the above-introduced unified multi-scale model. In particular the unary term, reflecting the similarity between the observation o_l^i and the hidden depth value d_i^l, is:

$$\phi(y_i^l, \mathbf{o}^l) = \big(d_i^l - o_i^l\big)^2, \tag{4.10}$$

where o_i^l is obtained via combining the regressed depth from the side output $\mathbf{s}^l$ and the map $\mathbf{d}^{l-1}$ estimated by the CRF at previous scale. In our implementation we simply consider $o_i^l = s_i^l + \tilde{d}_i^{l-1}$, but other alternative strategies can be also considered. The pairwise potentials, used to force neighboring pixels with similar appearance to have close depth values, are:

$$\psi(d_i^l, d_j^l) = \sum_{m=1}^{M}\beta_m K_m^{ij}(d_i^l - d_j^l)^2, \tag{4.11}$$

where we consider $M = 2$ Gaussian kernels, one for appearance features, and the other accounting for pixel positions. Similar to the multi-scale CRF model, under mean-field approximation, the following updates can be derived:

$$\gamma_{i,l} = 2\big(1 + 2\sum_{m=1}^{M}\beta_m \sum_{j\neq i} K_m^{ij}\big), \tag{4.12}$$

$$\mu_{i,l} = \frac{2}{\gamma_{i,l}}(o_i^l + 2\sum_{m=1}^{M}\beta_m \sum_{j\neq i} K_m^{ij}\mu_{j,l}). \tag{4.13}$$

At the test time, we use the estimated depth variables corresponding to the cascade CRF model of the finest scale L as our final predicted depth map $\mathbf{d}^\star$.

4.4 Multi-Scale Models as Sequential Deep Networks

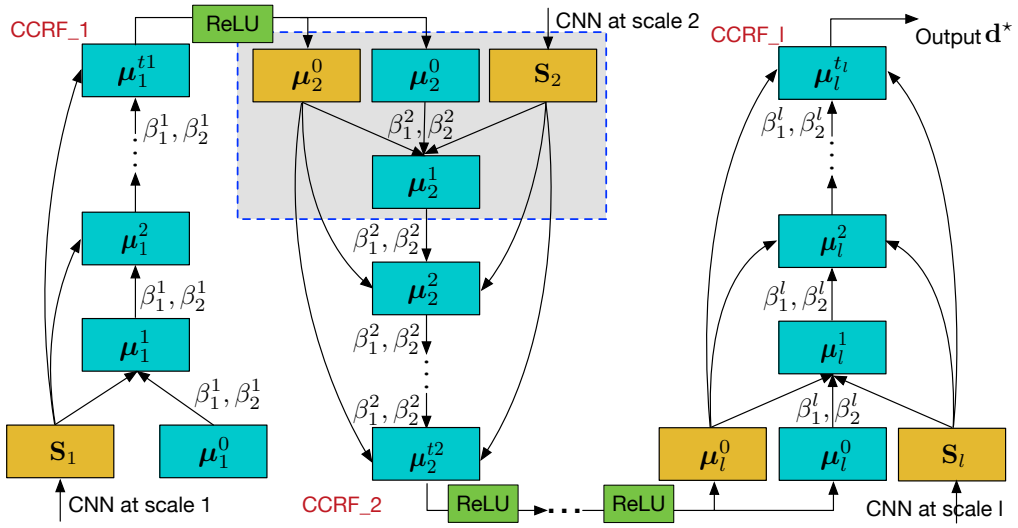
In this section, we describe how the two proposed CRFs-based models can be implemented as sequential deep networks, enabling end-to-end training of our whole deep network model (the front-end CNN and the fusion module). We first show how the mean-field iterations derived for the multi-scale and the cascade models can be implemented by designing a common structure, the continuous mean-field updating (C-MF) block, consisting into stack of a series of CNN operations. Then, we present the resulting sequential network structures and details of the training phase for optimizing the whole deep network.

4.4.1 C-MF: a Common CNN Implementation of Continuous Mean-Field Updating

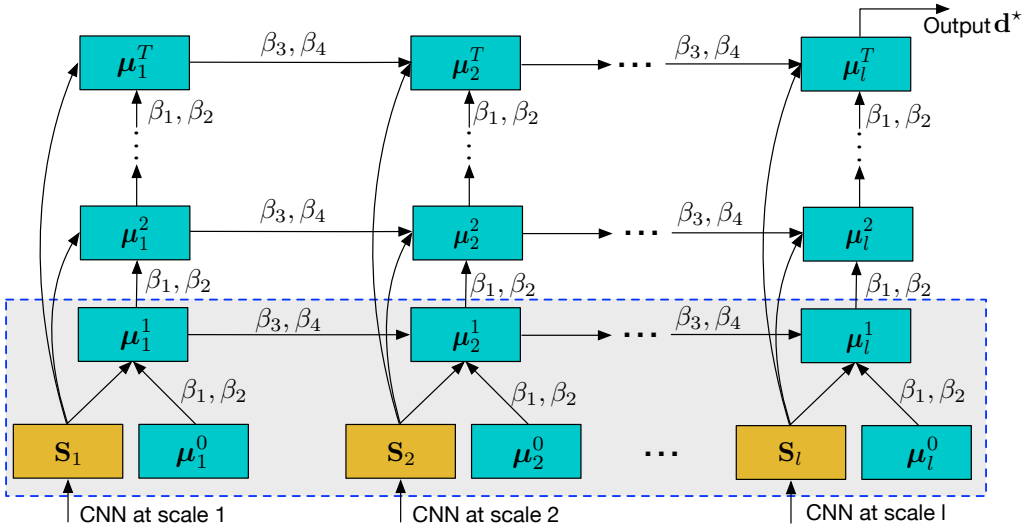
By analyzing the two proposed CRF models, we can observe that the mean-field updates derived for the cascade and for the multi-scale models share common terms. As stated above, the main difference between the two is the way the estimated depth at previous scale is handled at the current scale. In the multi-scale CRFs, the relationship among neighboring scales is modeled in the hidden variable space, while in the cascade CRFs the depth estimated at previous scale acts as an observed variable.

Starting from this observation, in this section we show how the computation of Eq. (4.8) and Eq. (4.13) can be implemented with a common structure. Figure 4.4 describes in details these computations. In the following, for the sake of clarity, we introduce matrix representation. Let $\mathbf{S}_l \in \mathbb{R}^{W\times H}$ be the matrix obtained by rearranging the $N = W \times H$ pixels corresponding to the side output vector $\mathbf{s}_l$ and $\boldsymbol{\mu}_l^t \in \mathbb{R}^{W\times H}$ the matrix of the estimated output depth variables associated to scale l and mean-field iteration t. To implement the multi-scale model at each iteration t, $\boldsymbol{\mu}_l^{t-1}$ and $\boldsymbol{\mu}_{l-1}^t$ are convolved by two Gaussian kernels. Following [68], we use a spatial and a bilateral kernel. As Gaussian convolutions represent the computational bottleneck (requiring a complexity of $\mathcal{O}(N^2)$) in the mean-field iterations, we adopt the permutohedral lattice implementation [1] to approximate the filter response calculation reducing the computational cost from quadratic to linear [118]. The weighing of the parameters β_m is performed as a convolution with a 1×1 kernel. Then, the outputs are combined and are added to the side-output maps $\mathbf{S}_l$. Finally, a normalization step follows, corresponding to the calculation of Eq. (4.7). The normalization matrix $\boldsymbol{\gamma} \in \mathbb{R}^{W\times H}$ is also computed by considering convolutions with Gaussian kernels and weighting with parameters β_m. It is worth noting that the normalization step in our mean-field updates for continuous CRFs is substantially different from that of discrete CRFs in CRF-RNN [188] based on a softmax function.

In the cascade CRF model, differently from the multi-scale unified CRF model, $\boldsymbol{\mu}_{l-1}^t$ acts as an observed variable. To design a common C-MF block among the two models, we introduce two gate functions G1 and G2 (Fig. 4.4) controlling the computing flow and allowing to easily switch between the two approaches. Both gate functions accept a user-defined boolean parameter. In our setting, the value 1 corresponds to the multi-scale CRF and the value 0 corresponds to the cascade model. Specifically, if G1 is equal to 1,



(a) The proposed multi-scale cascade CRF model as sequential neural network using the C-MF block.



(b) The proposed multi-scale unified CRF model as sequential neural network using the C-MF block.

Figure 4.5: Description of the proposed two CRF models as sequential deep networks. The blue and yellow boxes indicate the estimated variables and observations, respectively. The parameters β_m are used for mean-field updates. As in the cascade model parameters are not shared among different CRFs, we use the notation β_1^l, β_2^l to denote parameters associated to the l-th scale.

the gate function G1 passes $\boldsymbol{\mu}_{l-1}^{t}$ to the Gaussian filtering block, otherwise passes it to the element-wise addition block with the computed message. Similarly, G2 controls the computation of the normalization terms and switches between the computation of Eq. (4.7) and Eq. (4.12). In other words, if G2 equals to 0, then the Gaussian filtering and weighting operations for γ_3 and γ_4 are disabled. Importantly, for each step in the C-MF block we implement the calculation of error differentials for the back-propogation as in [188].

There are two different types of CRF parameters to be learned, *i.e.* the bandwidth parameters θ_m and the Gaussian-kernel weights β_m. For optimizing these CRF parameters, similar to [68], the bandwidth values θ_m are pre-defined for simplifying the calculation, and we implement the backward differential computation for the weights of Gaussian kernels β_m. In this way β_m are learned automatically with back-propagation.

4.4.2 From Mean-Field Updates to Sequential Deep Networks

Fig. 4.4 illustrates the implementation of the proposed two CRF-based models using the designed C-MF block described above. In the figure, each blue-dashed box is associated to a mean-field iteration. The cascade model as shown in Fig. 4.5(b) consists of L single-scale CRFs. At the l-th scale, t_l mean-field iterations are performed and then the estimated depth outputs are passed to another CRF model of the subsequent scale after a Rectified Linear Unit (ReLU) operation. The ReLU used here has two aspects of consideration: first the depth predictions should be always positive, and second we want to increase the nonlinearity of the sequential network for better mapping. To implement a single-scale CRF, we stack t_l C-MF blocks and make them share the parameters, while we learn different parameters for different CRFs. For the multi-scale model, one full mean-field update involves L scales simultaneously, obtained by combining L C-MF blocks. We further stack T iterations for learning and inference. The parameters corresponding to different scales and different mean-field iterations are shared. In this way, by using the common C-MF layer, we implement the two proposed multi-scale continuous CRFs models as deep sequential networks enabling end-to-end training with the front-end network.

4.4.3 Multi-Scale Message Passing Structures

The proposed work aims at multi-scale structured fusion and prediction, the connection structure between the different multi-scale predictions for message passing plays an important role in the performance. In this section, we thus propose and investigate different message passing structures. Fig. 4.3 illustrates several structures include top down structure, skip-connection structure and all to one structure. The top down structure is similar to the bottom up structure depicted in Fig. 5.2, which gradually refines the score maps from coarse to fine. The skip connection structure aims at utilizing more complementary information via skipping scales. The all to one structure uses all the other scales to refine the finest scale. Since all the message passing structures involve two scales at each time, we are able to build all these proposed connection structures by using the proposed aforementioned neural-network implemented C-MF block. The experimental investigation of these structures is illustrated in the experimental part.

4.4.4 Optimization of the Whole Network

We train the whole network using a two phase scheme. In the first phase (pretraining), the parameters of the base front-end network $\boldsymbol{\Theta}$ and the parameters of the side-output generation sub-branch networks $\boldsymbol{\vartheta} = \{\boldsymbol{\theta}_l\}_{l=1}^{L}$ are learned by minimizing the sum of L distinct side losses as in [158], corresponding to L side outputs. We define the optimization objective

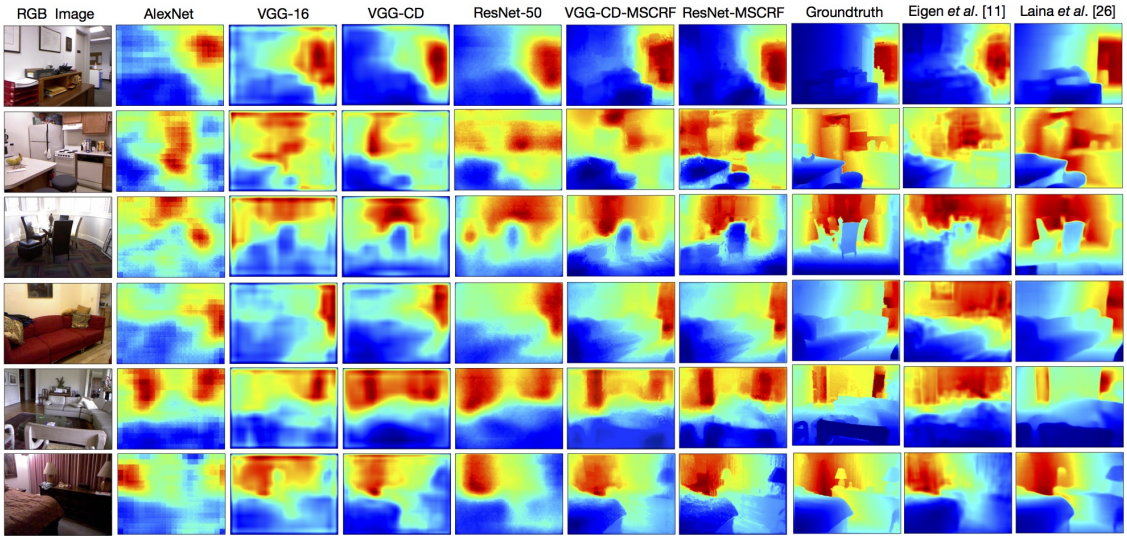


Figure 4.6: Examples of qualitative depth prediction results on the NYU v2 test dataset. Different front-end deep network architectures are investigated. VGG-CD-MSCRF and ResNet-MSCRF represent our approach with the proposed multi-scale continuous CRF model plugged on VGG-CD and ResNet-50 network respectively.

using a square loss over Q training samples as follows:

$$\{\boldsymbol{\Theta}^*, \boldsymbol{\vartheta}^*\} = \arg\min_{\boldsymbol{\Theta}, \boldsymbol{\theta}_l} \sum_{l=1}^{L} \sum_{i=1}^{Q} \| f_s(\mathbf{r}_i; \boldsymbol{\Theta}, \boldsymbol{\theta}_l) - \tilde{\mathbf{d}}_i \|_2^2, \tag{4.14}$$

where $\tilde{\mathbf{d}}_i$ denotes the i-th ground-truth sample. In the second phase (fine tuning), we initialize the front-end network with the learned parameters $\{\boldsymbol{\Theta}^*, \boldsymbol{\vartheta}^*\}$ in the first phase, and jointly fine-tune with the proposed multi-scale CRF models to compute the optimal value of the parameters $\boldsymbol{\Theta}$, $\boldsymbol{\vartheta}$ and $\boldsymbol{\beta}$, with $\boldsymbol{\beta} = \{\beta_m\}_{m=1}^{M}$. The entire network is learned with Stochastic Gradient Descent (SGD) by minimizing a square loss

$$\{\boldsymbol{\Theta}^*, \boldsymbol{\vartheta}^*, \boldsymbol{\beta}^*\} = \arg\min_{\boldsymbol{\Theta}, \boldsymbol{\vartheta}, \boldsymbol{\beta}} \sum_{i=1}^{Q} \| F(\mathbf{r}_i; \boldsymbol{\Theta}, \boldsymbol{\vartheta}, \boldsymbol{\beta}) - \tilde{\mathbf{d}}_i \|_2^2. \tag{4.15}$$

When the whole network optimization is finished, the test can be performed end-to-end, *i.e.* given a test RGB image as input the network directly outputs an estimated depth map.

4.5 Experiments

To demonstrate the effectiveness of the proposed multi-scale CRF models for monocular depth prediction, we performed experiments on three publicly available datasets: the NYU Depth V2 [132], the Make3D [125] and the KITTI [38] datasets. In the following we first describe the experimental setup and the implementation details, and then present the experimental results and analysis.

4.5.1 Experimental Setup

Datasets. The **NYU Depth V2** dataset [132] contains 120K unique pairs of RGB and depth images captured with a Microsoft Kinect. The datasets consists of 249 scenes for training and 215 scenes for testing. The images have a resolution of 640×480. To speed up the training phase, following previous works [89, 190] we consider only a small subset of images. This subset has 1449 aligned RGB-depth pairs: 795 pairs are used for training, 654 for testing. Following [30], we perform data augmentation for the training samples. The RGB and depth images are scaled with a ratio $\rho \in \{1, 1.2, 1.5\}$ and the depths are

Table 4.1: The parameter details of the sub-network for generating the side output from the last-scale convolutional block of ResNet-50.

Name	conv_s5_1	deconv_s5_1	deconv_s5_2
Type	conv	deconv	deconv
Kernel	$3\times3\times1024$	$4\times4\times512$	$4\times4\times256$
Stride, Padding	1, 1	2, 1	2, 1
Activation	ReLU	ReLU	ReLU
Name	deconv_s5_3	deconv_s5_4	pred
Type	deconv	deconv	deconv & crop
Kernel	$4\times4\times128$	$4\times4\times64$	$4\times4\times1$
Stride, Padding	2, 1	2, 1	2, 1
Activation	ReLU	ReLU	-

divided by ρ. Additionally, we horizontally flip all the samples and randomly crop them to 320×240 pixels. The data augmentation phase produces 4770 training pairs in total.

The **Make3D** dataset [125] contains 534 RGB-depth pairs, split into 400 pairs for training and 134 for testing. We resize all the images to a resolution of 460×345 as done in [92] to preserve the aspect ratio of the original images. We adopted the same data augmentation scheme used for NYU Depth V2 dataset but, for $\rho=\{1.2,1.5\}$ we randomly generate two samples each via cropping, obtaining 4K training samples.

The **KITTI** dataset [38] is built for various computer vision tasks within the context of autonomous driving, which contains depth videos captured through a LiDAR sensor deployed on a driving vehicle. For the training and testing split, we follow the protocol made by Eigen *et al.* [30] for a better comparison with existing works. Specifically, 61 scenes are selected from the raw data. Total 22,600 images from 32 scenes are used for training, and 697 images from the other 29 scenes are used for testing. Following [37], the ground-truth depth maps are generated by reprojecting the 3D points collected from velodyne laser into the left monocular camera. The resolution of RGB images are reduced half from original 1224×368 for training and testing.

Evaluation Metrics. Following previous works [29, 30, 153], we adopt the following evaluation metrics to quantitatively assess the performance of our depth prediction model. Specifically, we consider:

- mean relative error (rel): $\frac{1}{P}\sum_{i=1}^{P}\frac{|\tilde{d}_i-d_i^\star|}{d_i^\star}$;
- root mean squared error (rms): $\sqrt{\frac{1}{P}\sum_{i=1}^{P}(\tilde{d}_i-d_i^\star)^2}$;
- mean log10 error (log10):
  $\frac{1}{P}\sum_{i=1}^{P}\|\log_{10}(\tilde{d}_i)-\log_{10}(d_i^\star)\|$;
- scale invariant rms log error as used in [30], rms(sc-inv.);
- accuracy with threshold t: percentage (%) of $d_i^\star$,
  subject to $\max(\frac{d_i^\star}{\tilde{d}_i},\frac{\tilde{d}_i}{d_i^\star})=\delta<t$ $(t\in[1.25,1.25^2,1.25^3])$.

Where $\tilde{d}_i$ and $d_i^\star$ is the ground-truth depth and the estimated depth at pixel i respectively; P is the total number of pixels of the test images.

4.5.2 Implementation Details

We implemented the proposed deep model using the popular Caffe framework [15] on a single Nvidia Tesla K80 GPU with 12 GB memory. More details on the front-end CNN

Table 4.2: Quantitative performance comparison of different front-end deep network architectures and the proposed two multi-scale CRF models associated with the pretrained front-end networks on the NYU Depth V2 dataset.

Network Architecture	Error (lower is better)			Accuracy (higher is better)		
	rel	log10	rms	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
AlexNet (pretrain)	0.265	0.120	0.945	0.544	0.835	0.948
VGG-16 (pretrain)	0.228	0.104	0.836	0.596	0.863	0.954
VGG-ED (pretrain)	0.208	0.089	0.788	0.645	0.906	0.978
VGG-CD (pretrain)	0.203	0.087	0.774	0.652	0.909	0.979
ResNet-50 (pretrain)	0.168	0.072	0.701	0.741	0.932	0.981
AlexNet + cascade-CRFs	0.231	0.105	0.868	0.591	0.859	0.952
VGG-16 + cascade-CRFs	0.193	0.092	0.792	0.636	0.896	0.972
VGG-ED + cascade-CRFs	0.173	0.073	0.685	0.693	0.921	0.981
VGG-CD + cascade-CRFs	0.169	0.071	0.673	0.698	0.923	0.981
ResNet-50 + cascade-CRFs	**0.143**	**0.065**	**0.613**	**0.789**	**0.946**	**0.984**

Table 4.3: Quantitative baseline comparison with different multi-scale fusion schemes, and with the continuous CRF as a post-processing module on the NYU Depth V2 dataset. The number of scales is investigated for both multi-scale models with a bottom up message passing structure.

Method	Error (lower is better)			Accuracy (higher is better)		
	rel	log10	rms	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
HED [158]	0.185	0.077	0.723	0.678	0.918	0.980
Hypercolumn [49]	0.189	0.080	0.730	0.667	0.911	0.978
C-CRF	0.193	0.082	0.742	0.662	0.909	0.976
Ours (single-scale)	0.187	0.079	0.727	0.674	0.916	0.980
Ours - cascade (3-scale)	0.176	0.074	0.695	0.689	0.920	0.980
Ours - cascade (5-scale)	0.169	0.071	0.673	0.698	0.923	**0.981**
Ours - unified (3-scale)	0.172	0.072	0.683	0.691	0.922	**0.981**
Ours - unified (5-scale)	**0.163**	**0.069**	**0.655**	**0.706**	**0.925**	**0.981**

architectures, the generation of multi-scale side outputs and the parameter settings are elaborated as follows.

Front-end CNN Architectures. To study the influence of the frond-end CNN, we consider several network architectures including: (i) AlexNet [71], (ii) VGG-16 [133], (iii) a fully convolutional encoder-decoder network derived from VGG-16, referred as VGG-ED [5], (iv) a Convolution-Deconvolution network based on VGG-16, referred as VGG-CD [104], and (v) ResNet-50 [50]. For AlexNet, VGG-16 and ResNet-50, we obtain the side outputs from the last semantic convolutional layer of different convolutional blocks, in which each the layer produces feature maps with the same shape. The scheme utilized for the generation will be introduced in the next section. The number of side outputs considered in our experiments is 5, 5 and 4 for AlexNet, VGG-16 and ResNet-50, respectively. As VGG-ED and VGG-CD have been widely used for dense pixel-level prediction tasks, we also investigate them in the experimental analysis. Both VGG-ED and VGG-CD have a symmetric network structure, and five side outputs are then generated from the different blocks of the decoder or the deconvolutional network part.

Implementation details of CNN side outputs generation. Our approach can be applied with any multi-scale front-end CNN models including those with skip-connections. We here briefly describe the scheme we adopt to build CNN side outputs from the front-end CNN for the multi-scale fusion with CRFs. In [158] a convolutional layer is first used to

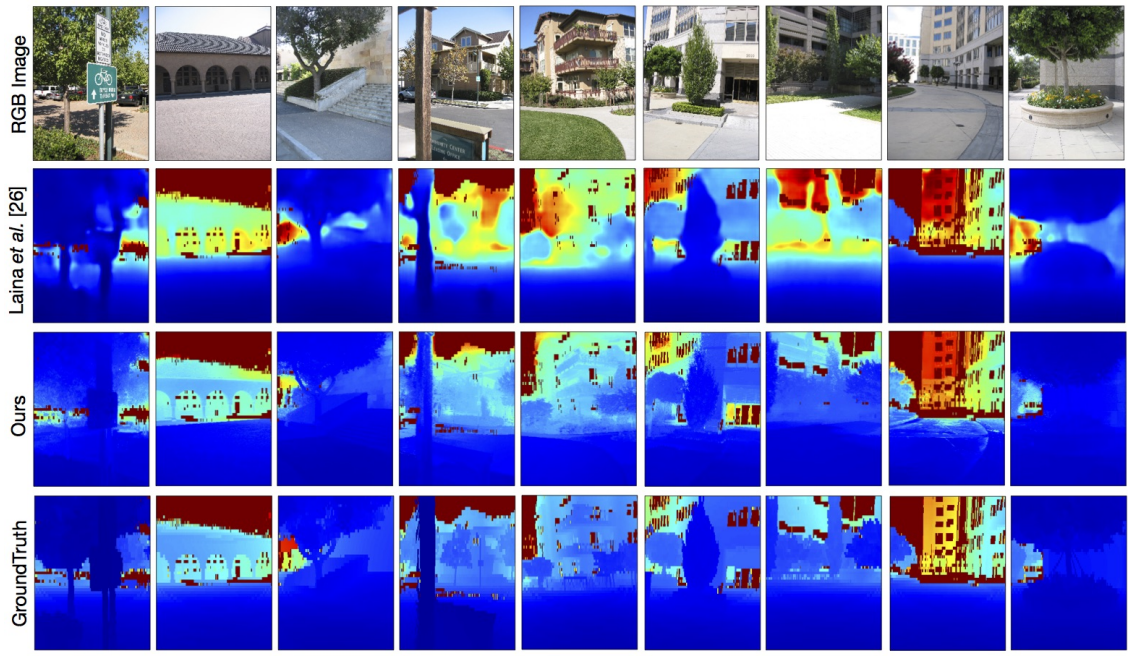


Figure 4.7: Examples of depth prediction results on the Make3D dataset. The four rows from up to bottom are the input test RGB images, the results produced from Laina *et al.* [77], the results of our ResNet50-MSCRF model and the groundtruth depth maps, respectively.

Table 4.4: Quantitative performance evaluation of different message passing structures for the cascade CRF model via building the sequential deep network with the proposed C-MF block on the NYU Depth V2 dataset.

Method	Error (lower is better)			Accuracy (higher is better)		
	rel	log10	rms	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Top down structure	0.175	0.072	0.688	0.689	0.919	0.979
Bottom up structure	0.169	0.071	0.673	0.698	0.923	0.981
Skip connection structure	0.161	0.070	0.664	0.709	0.923	0.981
All to one structure	**0.154**	**0.068**	**0.648**	**0.725**	**0.927**	**0.981**

generate a score map from the feature map and then a deconvolutional (*deconv*) layer is adopted as a bilateral upsampling operator to enlarge the score map such as to obtain the same size of the input image. However, we noticed that by adopting the approach in [158] the generated side outputs associated to the feature maps with smaller size are very coarse, causing a lot scene details missing. To address this problem, after the convolutional layer, we stack several *deconv* layers, each of them enlarging the output map by two times. A Rectified Linear Unit (ReLU) is applied after each *deconv* layer. After the last deconv layer we use a crop layer to cut the extra margin and obtain a side output with the same resolution of the ground-truth image. We employ this scheme to obtain side outputs for AlexNet, VGG-16 and ResNet-50, while for VGG-CD and VGG-ED, we use the same setting as in [158], as their decoder or deconvolutional part is able to obtain more fine-grained side outputs. Table 4.1 shows detailed network parameters used to obtain the side output from the last convolutional block of ResNet-50 (*i.e.* from the layer *res5c*).

Parameters settings. As described in Section 4.4.4, training consists of a pretraining and a fine tuning phase. In the first phase, we train the front-end CNN with parameters initialized with the corresponding ImageNet pretrained models. For AlexNet, VGG-16, VGG-ED and VGG-CD, the batch size is set to 12 and for ResNet-50 to 8. The learning

Table 4.5: Overall performance comparison with state of the art methods on the NYU Depth V2 dataset. Our approach achieves the best on most of the metrics, while the runners-up Eigen and Fergus [29] and Laina *et al.* [77] employ more training data than ours. ResNet-50-unified means using ResNet-50 front-end network with the proposed multi-scale unified CRF model.

Method	Error (lower is better)				Accuracy (higher is better)		
	rel	log10	rms	rms (sc-inv.)	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Karsch *et al.* [127]	0.349	-	1.214	0.325	0.447	0.745	0.897
Ladicky *et al.* [64]	0.35	0.131	1.20	-	-	-	-
Liu *et al.* [92]	0.335	0.127	1.06	-	-	-	-
Ladicky *et al.* [75]	-	-	-	-	0.542	0.829	0.941
Zhuo *et al.* [190]	0.305	0.122	1.04	-	0.525	0.838	0.962
Liu *et al.* [89]	0.230	0.095	0.824	-	0.614	0.883	0.975
Wang *et al.* [153]	0.220	0.094	0.745	-	0.605	0.890	0.970
Eigen *et al.* [30]	0.215	-	0.907	0.219	0.611	0.887	0.971
Roi and Todorovic [119]	0.187	0.078	0.744	-	-	-	-
Eigen and Fergus [29]	0.158	-	0.641	0.171	0.769	0.950	0.988
Laina *et al.* [77]	0.129	0.056	0.583	-	0.801	0.950	0.986
Ours (ResNet-50-unified-4.7K-bottom up)	0.139	0.063	0.609	0.163	0.793	0.948	0.984
Ours (ResNet-50-unified-95K-bottom up)	0.121	0.052	0.586	0.149	0.811	0.954	0.987
Ours (ResNet-50-unified-95K-all to one)	**0.108**	**0.045**	**0.579**	**0.142**	**0.823**	**0.957**	**0.987**

rate is initialized at 10^{-11} and decreases by 10 times around every 50 epochs. 80 epochs are performed for pretraining in total. The momentum and the weight decay are set to 0.9 and 0.0005, respectively. When the pretraining is finished, we connect all the side outputs of the front-end CNN to our CRFs-based multi-scale deep models for end-to-end training of the whole network. In this phase, the batch size is reduced to 6 and a fixed learning rate of 10^{-12} is used. The same parameters of the pre-training phase are used for momentum and weight decay. The bandwidth weights for the Gaussian kernels are obtained through cross validation. The number of mean-field iterations is set to 5 for efficient training for both the cascade CRFs and multi-scale CRFs. We do not observe significant improvement using more than 5 iterations. Training the whole network takes around ~ 25 hours on the Make3D dataset, ~ 28 hours on the KITTI dataset and ~ 31 hours on the NYU v2 dataset.

4.5.3 Experimental Results

To present the experimental results, we start from an ablation study for investigating the performance impact of different front-end network architectures, the effectiveness of the proposed CRF-based multi-scale fusion models and the influence of the stacking orders for making the sequential neural network. Then we compare the overall performance with the state of the art methods, and finally the qualitative results and running time are analyzed.

Evaluation of different front-end CNN architectures. As discussed above, the proposed multi-scale CRF-based fusion models are general and different deep architectures can be used for the front-end network. In this section we evaluate the impact of this choice on the depth estimation performance. We consider both the case of the pretrained front-end models (*i.e.* only side losses are employed but the multi-scale CRF models are not plugged), indicated with 'pretrain', and the case of the fine-tuned models, including the front-end network with the multi-scale cascade CRFs (cascade-CRFs). The results of the experiments are shown in Table 4.2. As expected, in both cases deeper CNN architectures produced more accurate predictions, and ResNet-50 achieves the best performance among all the front-end networks. Moreover, VGG-CD is slightly better than VGG-ED, and both these models outperforms VGG-16, showing that the symmetric network structure is beneficial for the dense pixel-level prediction problems. Importantly, for all considered

Table 4.6: Overall performance comparison with state of the art methods on the Make3D dataset. Our approach outperforms all the competitors w.r.t. the C2 Error, and performs only slightly worse on the *rel* metric of the C1 Error than Laina *et al.* [77] using Huber loss and significantly larger training data.

Method	C1 Error				C2 Error		
	rel	log10	rms	rms (sc-inv.)	rel	log10	rms
Karsch *et al.* [64]	0.355	0.127	9.20	-	0.361	0.148	15.10
Liu et al. [92]	0.335	0.137	9.49	-	0.338	0.134	12.60
Liu et al. [89]	0.314	0.119	8.60	-	0.307	0.125	12.89
Li et al. [82]	0.278	0.092	7.19	-	0.279	0.102	10.27
Laina *et al.* [77] (ℓ_2 loss)	0.223	0.089	4.89	-	-	-	-
Laina *et al.* [77] (Huber loss)	0.176	0.072	4.46	-	-	-	-
Ours (ResNet-50-cascade-bottom up)	0.213	0.082	4.67	0.245	0.221	4.79	8.81
Ours (ResNet-50-unified-bottom up)	0.206	0.076	4.51	0.237	0.212	4.71	8.73
Ours (ResNet-50-unified-10K-bottom up)	0.184	0.065	4.38	0.219	0.198	4.53	8.56
Ours (ResNet-50-unified-10K-all to one)	**0.174**	**0.059**	**4.27**	**0.211**	**0.185**	**4.41**	**8.43**

front-end networks there is a significant increase in performance when applying the proposed CRF-based models.

Figure 5.4 depicts some examples of predicted depth maps using different front-end networks on the NYU Depth V2 test dataset. As we can see from the figure, the qualitative results confirm that the deeper architecture leads to better depth recovery. By comparing the reconstructed depth maps obtained with pretrained models (*e.g.* using only the front-end networks VGG-CD and ResNet-50) with those generated with our multi-scale models, it is clear that our approach remarkably improves prediction accuracy and visual quality.

Evaluation of different multi-scale CRF fusion models. To evaluate the effectiveness of the proposed CRF-based multi-scale fusion models, we conduct experiments on the NYU Depth V2 dataset and consider the following baselines:

(i) the 'HED' method in [158], where multiple side outputs are fused with a weighted averaging scheme and the sum of multiple side output losses is jointly minimized as deep supervision with a cross-entropy loss, while we use the square loss as our problem involves continuous variables;

(ii) the 'Hypercolumn' method [49], where multi-scale feature maps generated from different semantic network layers are concatenated and fused;

(iii) a continuous CRF ('C-CRF') applied on the prediction of the front-end network, *i.e.* plugging after the last output layer as a post-processing module without end-to-end training.

For the first two baselines, we want to compare our models with other popular methods for fusing multi-scale CNN information, while the third one aims at demonstrating the effectiveness of the continous CRF itself. In these experiments we consider VGG-CD as the front-end CNN architecture. The results of the comparison are shown in Table 4.3. It is evident that with our CRF-based fusion models (both the cascade CRFs and the unified CRFs) more accurate depth maps can be obtained, demonstrating that our idea of integrating complementary information derived from CNN side output maps within a graphical model framework is more effective than traditional fusion schemes. Table 4.3 also compares the proposed cascade and unified models. As expected, the unified model produces more accurate depth maps, at the price of an increased computational cost. This can also be observed from Table 4.2. The C-CRF (in Table 4.3) improves the depth estimation at all metrics over the VGG-CD (pretrain) (in Table 4.2) with a clear gap,

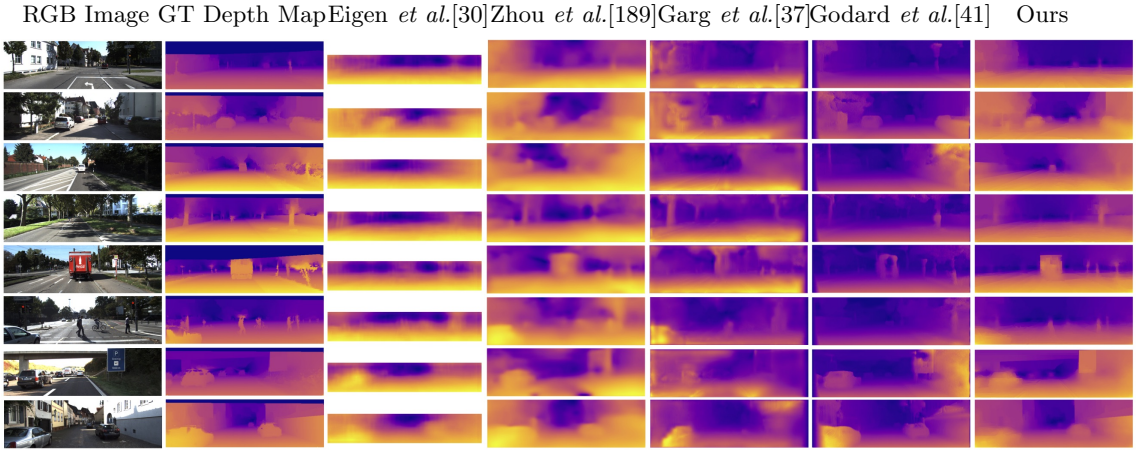


Figure 4.8: Examples of depth prediction results on the KITTI raw dataset. Qualitative comparison with other depth estimation methods on this dataset is presented. The sparse ground-truth depth maps are interpolated for better visualization.

Table 4.7: Overall performance comparison with state of the art methods on the KITTI raw dataset. Our approach obtains very competitive performance over all the competitors w.r.t. all the evaluation metrics on the testing set given by Eigen *et al.* [30]. For the setting, caps means different gt/predicted depth range and stereo means using left and right images captured from two monocular cameras in the training phase. Ours uses a unified model considering both the bottom up and the all to one network structure.

Method	Setting		Error (lower is better)				Accuracy (higher is better)		
	range	stereo	rel	sq rel	rms	rms (sc-inv.)	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Saxena *et al.* [127]	0-80m	No	0.280	-	8.734	0.327	0.601	0.820	0.926
Eigen *et al.* [30]	0-80m	No	0.190	-	7.156	0.246	0.692	0.899	0.967
Liu *et al.* [89]	0-80m	No	0.217	0.092	7.046	-	0.656	0.881	0.958
Zhou *et al.* [189]	0-80m	No	0.208	1.768	6.858	-	0.678	0.885	0.957
Kuznietsov *et al.*[74] (only supervised)	0-80m	No	-	-	4.815	-	0.845	0.957	0.987
Garg *et al.* [37]	0-80m	Yes	0.177	1.169	5.285	-	0.727	0.896	0.962
Garg *et al.* [37] L12 + Aug 8x	1-50m	Yes	0.169	1.080	5.104	-	0.740	0.904	0.958
Godard *et al.* [41]	0-80m	Yes	0.148	1.344	5.927	-	0.803	0.922	0.964
Kuznietsov *et al.*[74]	0-80m	Yes	-	-	**4.621**	-	**0.852**	**0.960**	**0.986**
Ours (ResNet-50 Pretrain)	0-80m	No	0.152	0.973	4.902	0.176	0.782	0.931	0.975
Ours (ResNet-50 Fine-tune-bottom up)	0-80m	No	0.132	0.911	4.791	0.162	0.804	0.945	0.981
Ours (ResNet-50 Fine-tune-all to one)	0-80m	No	**0.125**	**0.899**	4.685	**0.154**	0.816	0.951	0.983

showing the CRF model is very useful for refining the deeply predicted map. By jointly learning with the front-end (*i.e.* end-to-end training), ours (single-scale) further boosts the performance. Finally, we analyze the impact of adopting multiple scales and compare our complete models (5 scales) with their version when only a single and three side output layers are used. It is evident that the performance can be improved by increasing the number of scales.

Evaluation of multi-scale message passing structures. We evaluate the influence of different multi-scale message passing structures using the cascade CRF model. Four connection structures as depicted in Fig. 4.3 are compared. Table 4.4 shows the monocular depth estimation results on NYUD-v2 dataset. The comparison results confirm that the message passing structure indeed has an impact on the final performance. The bottom up and top down structures have similar performance, while the skip-connection structure slightly outperform these two. The all to one structure performs the best, producing around 2.0% gain in terms of the *rel* metric than the top down structure, which means that directly passing message to the finest prediction scale from the rest scales can absorb more complementary information than the gradual passing fashions used in the first three

structures.

Comparison with state of the art. We also compare our approach with state of the art methods on all the datasets. For previous works we directly report results taken from the original papers. Table 4.5 shows the results of the comparison on the NYU Depth V2 dataset. For our approach we consider the cascade model and use two different training sets for pretraining: the small set of 4.7K pairs employed in all our experiments and a larger set of 95K images as in [77]. Note that for fine tuning we only use the small set. As shown in the table, our approach outperforms all competing methods and it is the second best model when we use only 4.7K images. This is remarkable considering that, for instance, in [29] 120K image pairs are used for training. Our model achieves the best results on all the metrics via using 95K pretraining samples and using the proposed all to one message passing structure.

We also perform a comparison with several state of the art methods on the Make3D dataset (Table 4.6). Following [92], the error metrics are computed in two different settings, *i.e.* considering (C1) only the regions with ground-truth depth less than 70 and (C2) the entire image. It is clear that the proposed approach is significantly better than previous methods. In particular, comparing with Laina *et al.* [77], the best performing method in the literature, it is evident that our approach, both in case of the cascade and the multi-scale models, outperforms [77] by a significant margin when Laina *et al.*also adopt a square loss. It is worth noting that in [77] a training set of 15K image pairs is considered, while we employ much less training samples. By increasing our training data (*i.e.* $\sim$ 10K in the pretraining phase), our multi-scale CRF model also outperforms [77] with Huber loss (log10 and rms metrics). The final performance is further boosted by considering the all to one structure similar to NYUD v2 dataset. Finally, it is very interesting to compare the proposed method with the approach in Liu *et al.*[89], since in [89] a CRF model is also employed within a deep network trained end-to-end. Our method significantly outperforms [89] in terms of accuracy. Moreover, in [89] a time of 1.1sec is reported for performing inference on a test image but the time required by superpixels calculations is not taken into account. Oppositely, with our method computing the depth map for a single image takes about 1 sec in total.

The state of the art comparison on KITTI dataset is shown in Table 4.7. The competitors include Saxena *et al.* [125], Eigen *et al.* [30], Liu *et al.* [90], Zhou *et al.* [189], Garg *et al.* [37], Godard *et al.* [41] and Kuznietsov *et al.*[74]. As the same setting of ours, the first four methods use single monocular images in the training phase, while the last two considered two monocular images with a stereo setting for training. Among the first four competitors, Eigen *et al.* [30] significantly outperforms the others in terms of the metric of the mean relative error (*rel*), due to the usage of large-scale training data (more than 1 million samples). While our model achieves much better performance than Eigen *et al.* [30] in all metrics with much less data (22.6K samples). Although the training of the last two methods (requiring two monocular images) is not equal to our setting, the proposed approach with both the bottom-up and the all to one structures still produces better results than them with clear performance gap in all metrics. Kuznietsov *et al.*[74] reports results for both the stereo training and the monocular supervised training. It is not directly comparable with the stereo training setting, which is significantly different as it requires both left and right images from a binocular camera. Ours focuses on monocular depth estimation and achieves lower error performance comparing with theirs using the

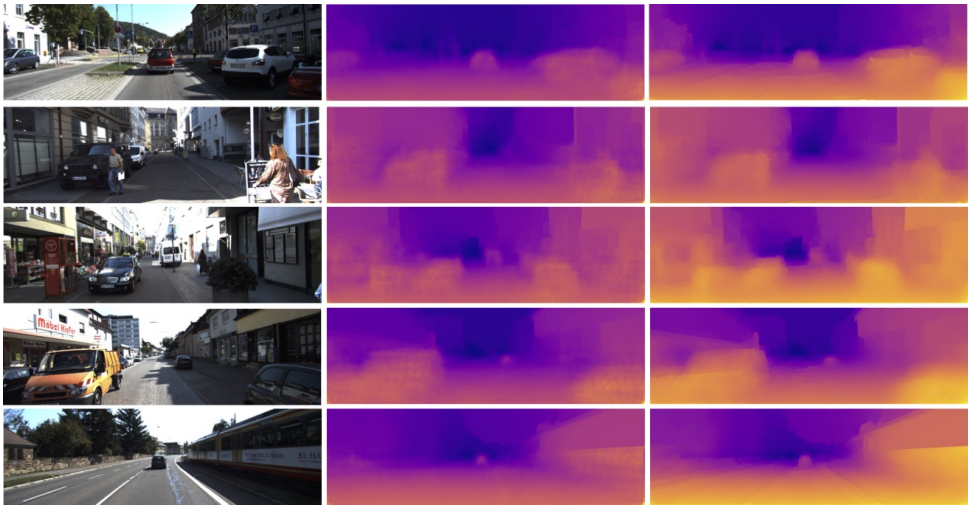


Figure 4.9: Examples of depth prediction results on the KITTI raw dataset. The middle column and the right column show the pretrained and the fine-tuned estimation results respectively.

same monocular setting. Fig. 4.8 also shows some qualitative comparison results with these methods, further demonstrating the advantageous performance of our approach.

Qualitative depth estimation results. Fig. 5.4, 4.7 and 4.9 show some examples of the qualitative depth estimation results and the comparison with the competing methods on the NYUD-V2, Make3D and KITTI dataset respectively. It is clear that the proposed approach is able to produce sharper depth estimation with better visual quality compared with the classic CNN structures, which demonstrates the importance of the prediction aided by the CRFs with appearance and smoothness constraints. Fig. 4.9 also shows a qualitative comparison between the pretrained front-end CNN and the fine-tuned whole model. It can be observed that our approach can recover more scene structures and details. We believe that this is probably because the effective structured fusion of the coarse-to-fine multi-scale predictions of the deep network with the proposed CRF models.

Empirical run-time analysis. Computational run-time complexity is an important aspect for deep structured prediction models. In this paragraph we provide a short discussion about the computational cost of the proposed CRFs-based models. As shown in the paper, the multi-scale CRF model achieves better accuracy and lower error than the cascade model for both the NYU Depth V2 and the Make3D experiments. However, as expected, the cascade model is more advantageous in terms of the running time. For instance, considering ResNet-50 as the front-end CNN, the time required at test phase for one image is 1.02 seconds w.r.t. the cascade model and 1.45 seconds w.r.t. the multi-scale model, and the image resolution is 320×240 pixels. Higher resolution of the network input usually brings more computational overhead. We also test the running time given the input resolution of 640×480 and it costs around 2.25 seconds for processing one image. We believe that if we reduce the receptive field of the CRF model from fully connected to partially connected, the computing time could be significantly reduced.

4.6 Conclusion

In this work, we introduced a novel approach for predicting depth maps from a single RGB image. The core of the method is a novel framework based on continuous CRFs for fusing multi-scale score-level side-outputs derived from different semantic CNN layers. We

demonstrated that this framework can be used in combination with several common CNN architectures and can be implemented for end-to-end training. The extensive experiments confirmed the validity of the proposed multi-scale fusion approach. While this paper specifically addresses the problem of depth prediction, we believe that other tasks in computer vision involving pixel-level predictions of continuous variables, can also benefit from our implementation of the mean-field updating within the CNN framework.

Currently, the multi-scale fusion is performed on the score level. Further research direction will investigate the integration of both the feature- and the score-level multi-scale information within a unified graphical model. Moreover, the study of strategies for further improving the training and testing efficiency of the CNN-CRF models will also be an interesting aspect in the future work. The monocular depth estimation is particularly useful for various cross-modal recognition and detection tasks. A straightforward follow-up of this work would be designing a joint multi-task deep model to transfer the learned depth model for aiding other similar dense prediction problems such as contour detection and semantic segmentation.

CHAPTER 5

Deep Multi-Modal Prediciton-and-Distillation for Simultaneous Depth Estimation and Scene Parsing [1]

Depth estimation and scene parsing are two particularly important tasks in visual scene understanding. In this chapter we tackle the problem of simultaneous depth estimation and scene parsing in a joint CNN. The task can be typically treated as a deep multi-task learning problem [116]. Different from previous methods directly optimizing multiple tasks given the input training data, this paper proposes a novel multi-task guided prediction-and-distillation network (PAD-Net), which first predicts a set of intermediate auxiliary tasks ranging from low levels to high levels, and then the predictions from these intermediate auxiliary tasks are utilized as input via our proposed multi-task distillation modules for the final tasks. During the joint learning, the intermediate tasks not only act as supervision for learning more robust deep representations but also provide rich multi-modal information for improving the final tasks. Extensive experiments are conducted on two challenging datasets (*i.e.* NYUD V2 and Cityscapes) for both the depth estimation and scene parsing tasks, demonstrating the effectiveness of the proposed approach.

5.1 Introduction

Depth estimation and scene parsing are both fundamental tasks for visual scene perception and understanding. Significant efforts have been made by many researchers on the two tasks in recent years. Due to the powerful deep learning technologies, the performance of the two individual tasks has been greatly improved [30, 164, 19]. Since these two tasks are correlated, jointly learning a single network for the two tasks is a promising research line.

Typical deep multi-task learning approaches mainly focused on the final prediction level via employing the cross-modal interactions to mutually refining the tasks [58, 153] or designing more effective joint optimization loss functions [101, 66]. These methods directly learn to predict the two tasks given the same input training data. Under this setting, it usually requires the deep model to share the network parameters or features. However, simultaneously learning different tasks using different loss functions makes the network

[1] Dan Xu, Wanli Ouyang, Xiaogang Wang, Nicu Sebe, "PAD-Net: Multi-Tasks Guided Prediction-and-Distillation Network for Simultaneous Depth Estimation and Scene Parsing", IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2018).

optimization complicated, and it is generally not easy to obtain a good generalization ability for both tasks.

In this work, we explore multi-task learning from a different direction, use the multi-task outputs as multi-modal input data. This is motivated by three observations. First, it is well-known that multi-modal data improves performance. Take the task of scene parsing as an example, a CNN with RGBD data should perform better than a CNN with only RGB. If we do not have the depth as the input, we can use the CNN to predict the depth map and then use it as the input. Second, instead of using the output from the target tasks, segmentation and depth, as the multi-modal input, we can use the power of CNN to predict more, *e.g.* contour, surface normal, *etc.* Third, instead of directly using multi-task output to refine each other, treating output from one task as the input of CNN facilitates transforming it to another task through multi-layer nonlinear operations. For example, it would be better to use multi-layer nonlinear operations to obtain/refine scene parsing results from depth instead of directly using depth to refine scene parsing results.

Based on the observations above, a multi-task guided prediction-and-distillation network (PAD-Net) is proposed. Specifically, we first learn to use a front-end deep CNN and the input RGB data for producing a set of intermediate auxiliary tasks, as shown in Figure 5.1. The auxiliary tasks range from low levels to high levels including two continuous regression tasks (depth prediction, surface normal estimation) and two discrete classification tasks (semantic parsing and contour detection). The produced multiple predictions, *i.e.* depth, surface normal, semantic labels and object contours, are then utilized as the multi-modal input of the next deep CNN for the final two main tasks. By involving an intermediate multi-task prediction module, the proposed PAD-Net not only adds deep supervision for optimizing the front-end network more effectively, but also is able to incorporate more knowledge from relevant domains. Since the predicted multi-modal results are highly relevant, we further propose multi-task distillation strategies to better using these data. When the learning of the whole PAD-Net is finished, the inference is only based on the RGB input.

To summarize, the contribution of this paper is threefold: (i) First, we propose a new multi-tasks guided prediction-and-distillation network (PAD-Net) structure for simultaneous depth estimation and scene parsing. It produces a set of intermediate auxiliary tasks as multi-modal data and then use them for the target tasks. Although PAD-Net takes only RGB data as input, it is able to incorporate multi-modal information for improving the final tasks. (ii) Second, we design and investigate three different multi-task distillation modules for deep multi-modal data fusion, which we believe can be also applied in other scenarios such as multi-scale deep feature fusion.

Extensive experiments the challenging NYUD v2 and Cityscapes datasets demonstrate the effectiveness of the proposed approach. We also show our approach achieves state-of-the-arts results on NYUD V2 on both the depth estimation and the scene parsing tasks and very competitive performance on the Cityscapes scene parsing task. More importantly, our method remarkably outperforms state-of-the-art methods working on jointly optimizing the two tasks. Our code will be made publicly available upon acceptance.

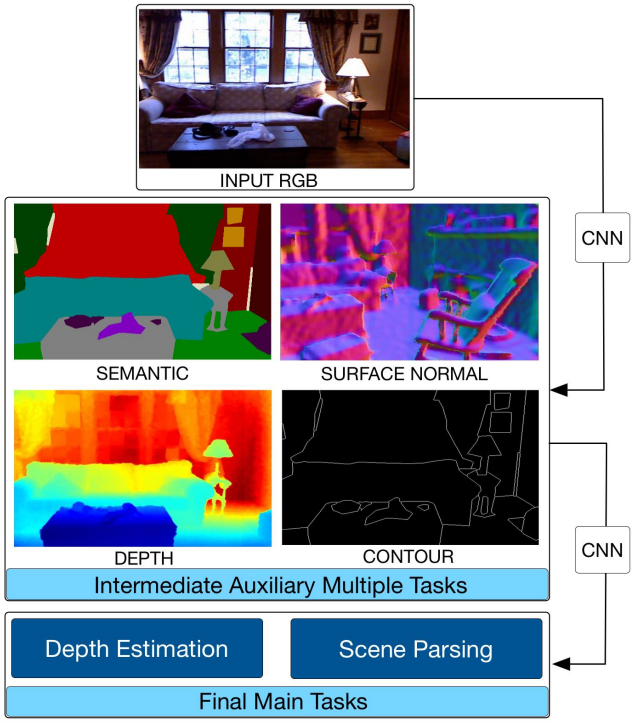


Figure 5.1: Illustration of the motivation that uses multiple intermediate multi-task predictions as guidance to facilitate the final main-tasks. Multiple intermediate tasks are considered, ranging from low levels to high levels including depth prediction, surface normal estimation, contour prediction and semantic parsing.

5.2 Related Work

5.2.1 Depth Estimation and Scene Parsing.

The works on monocular depth estimation can be mainly grouped into two categories. The first group comprises the methods based on the hand-crafted features and graphical models [25, 126, 92]. For instance, Saxena *et al.* [126] proposed a discriminatively-trained Markov Random Field (MRF) model for multi-scale estimation. Liu *et al.* [92] built a discrete and continuous Conditional Random Field (CRF) model for fusing both local and global features. The second group of the methods is based on the advanced deep learning models [29, 89, 153, 119, 77]. Eigen *et al.* [30] developed a multi-scale CNN for fusing both coarse and fine predictions from different semantic layers of the CNN. Recently, researchers studied implementing the CRF models with CNN enabling the end-to-end optimization of the whole deep network [89, 164].

Many efforts have been devoted to the scene parsing task in recent years. The scene parsing task is usually treated as a pixel-level prediction problem and the performance is greatly boosted by the fully convolutional strategy [94] which replaces the full connected layers with convolutional layers and dilated convolution [19, 177]. The other works mainly focused on multi-scale feature learning and ensembling [21, 156, 49], end-to-end structure prediction with CRF models [93, 4, 188] and designing convolutional encoder-decoder network structures [104, 5]. These works focused on an individual task but not jointly optimizing the depth estimation and scene parsing together.

Some works [101, 153, 58, 72] explored simultaneously learning the depth estimation and the scene parsing tasks. For instance, Wang *et al.* [153] introduced an approach to

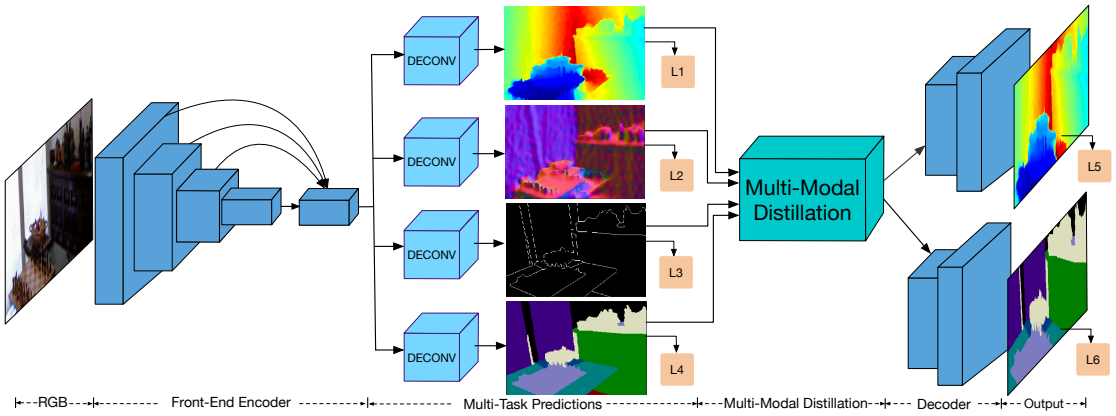


Figure 5.2: Illustration of the proposed multi-task distillation network for simultaneous depth estimation and scene parsing. The symbols from L1 to L6 denote different optimization losses for different tasks. 'DECONV' denotes the deconvolutional operation for upsampling and generating task-specific feature maps. The cube 'Multi-task Distillation' represents our proposed multi-task distillation module for fusing the multiple predictions to improve the final main tasks.

model the two tasks within a hierarchical CRF, while the CRF model is not jointly learned with the CNN. However, these works directly learn the two tasks without treating them as multi-modal input for the final tasks.

5.2.2 Deep Multi-task Learning for Vision.

Deep multi-task learning [95, 99, 120] has been widely used in various computer vision problems, such as joint inference scene geometric and semantic [66], face attribute estimation [47], simultaneous contour detection and semantic segmentation [44]. Yao and Urtasun *et al.* [176] proposed an approach for joint learning three tasks *i.e.* object detection, scene classification and semantic segmentation. Hariharan *et al.* [48] proposed to simultaneously learn object detection and semantic segmentation based on the R-CNN framework. However, none of them considered introducing intermediate multi-task prediction and distillation steps at the intermediate input level to for improving the target tasks.

5.3 PAD-Net: Multi-tasks Guided Prediction-and-Distillation Network

In this section, we describe the proposed PAD-Net for simultaneous depth estimation and scene parsing. We first present an overview of the proposed PAD-Net, and then introduce the details of the PAD-Net. Finally, we illustrate the optimization and inference schemes for the overall network.

5.3.1 Approach Overview

Figure 5.2 depicts the framework of the proposed multi-tasks guided prediction and distillation network (PAD-Net). PAD-Net consists of four main components. First, a front-end fully convolutional encoder produces deep features. Second, an intermediate multi-task prediction module, which uses the deep features in the previous component for generating intermediate predictions. Third, a multi-task distillation module which is used for incorporating useful information from the intermediate predictions to improve the final tasks. Fourth, the decoders uses the distilled information for depth estimation and scene parsing. The input of PAD-Net is RGB images during both training and testing, and the final output is the depth and semantic parsing maps. During training, labels of scene

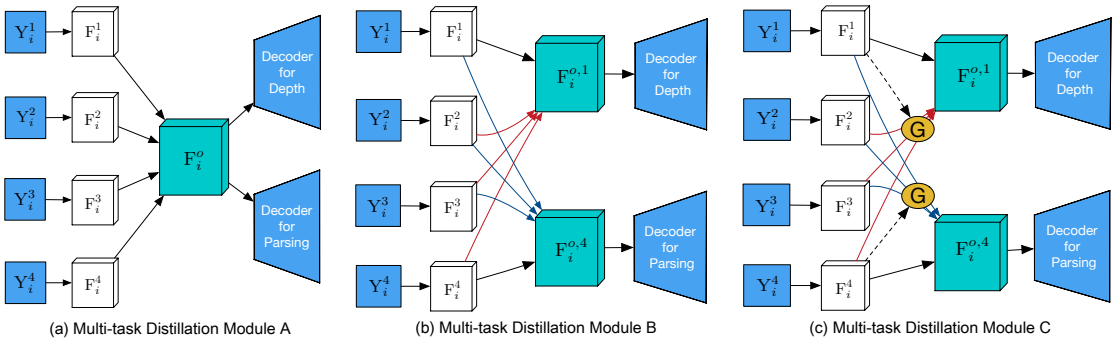


Figure 5.3: Illustration of the different multi-task distillation modules. The symbols Y_i^1, Y_i^2, Y_i^3, Y_i^4 represent the predictions corresponding to multiple intermediate tasks. The distillation module A is a naive combination of the multiple predictions; the module B proposes a mechanism of passing message between different predictions; the module C shows an attention-guided message passing mechanism for distillation. The symbol G denotes a generated attention map which is used as guidance in the distillation.

parsing, depth estimation and other intermediate tasks, *e.g.* surface normal and contour, are used.

5.3.2 Front-End Network Structure

The front-end backbone CNN could employ any network structures, such as the commonly used AlexNet [71], VGG [133] and ResNet [50]. To obtain better deep representations for predicting multiple intermediate tasks, we do not directly use the features from the last convolutional layer of the backbone CNN. A multi-scale feature aggregation procedure is performed to enhance the last-scale feature map via combining the previous scales feature maps derived from different semantic layers of the backbone CNN, as shown in Figure 5.2. The larger-resolution feature maps from shallower layers are down-sampled via convolution and bilinear interpolation operations to the resolution of the last-scale feature map. The convolution operations are also used to control the number of feature channels to make the feature aggregation more memory efficient. And then all the re-scaled feature maps are concatenated for the follow up deconvolutional operations. Similar to [18, 177], we also apply the dilated convolution strategy in the front-end network to produce feature maps with enlarged receptive field.

5.3.3 Deep Multi-task Prediction

Using deep features from the front-end CNN, we perform deconvolutional operations to generate four sets of task-specific feature maps. We obtain N feature channels for the main depth estimation and scene parsing tasks while $N/2$ feature channels for the other two auxiliary tasks. The feature map resolution is the same for four tasks and is $2\times$ as that of the front-end feature maps. Then different convolutional operations are used to produce the score maps for the four tasks. The score maps are made to be $1/4$ as the resolution of the input RGB images via the bilinear interpolation. Four different loss functions are added for learning the four intermediate tasks with the re-scaled groudtruth maps. It should be noted that the intermediate multi-task learning not only provides deep supervision for optimizing the front-end CNN, but also helps to provide valuable multi-modal predictions, which are used as input for the final tasks.

5.3.4 Deep Multi-task Distillation

As mentioned before, the deep multi-task distillation module fuses information from the intermediate predictions for the final tasks. It aims at effectively utilizing the complementary information from the intermediate predictions of relevant tasks. To achieve this goal, any distillation scheme can be used. In this work, we develop and investigate three different module designs as shown in Figure 5.3. The distillation module A represents a naive concatenation of the features extracted from these predictions. The distillation module B passes message between different predictions. The distillation module C is an attention-guided message passing mechanism for information fusion. To generate richer information and bridge the gap between these predictions, before the distillation procedure, all the intermediate prediction maps associated with the i-th training sample, denoted by $\{\mathrm{Y}_i^t\}_{t=1}^T$, are first correspondingly transformed to feature maps $\{\mathrm{F}_i^t\}_{t=1}^T$ with more channels via convolutional layers, where T is the number of intermediate tasks.

Multi-Task Distillation module A. A common way in deep networks for information fusion is to perform a naive concatenation of the feature maps or the score maps from different semantic layers of the network. We aslo consider this simple scheme as our basic distillation module. The module A outputs only one set of fused feature maps via $\mathrm{F}_i^o \leftarrow \mathrm{CONCAT}(\mathrm{F}_i^1, ..., \mathrm{F}_i^T)$, where $\mathrm{CONCAT}(\cdot)$ denotes the concatenation operation. And then F_i^o is fed into different decoders for predicting different final tasks, *i.e.* the depth estimation and the scene parsing tasks.

Multi-Task Distillation module B. The module A outputs the same set of feature maps for the two final tasks. Differently, the module B learns a separate set of feature maps for each final task. For the k-th final task, let us denote F_i^k as the feature maps before message passing and denote $\mathrm{F}_i^{o,k}$ as the feature maps after the distillation. We refine F_i^k via passing message from the feature maps of other tasks as follows:

$$\mathrm{F}_i^{o,k} \leftarrow \mathrm{F}_i^k + \sum_{t=1(\neq k)}^{T} (\mathrm{W}_{t,k} \otimes \mathrm{F}_i^t), \tag{5.1}$$

where $\otimes$ denotes convolution operation, and $\mathrm{W}_{t,k}$ denotes the convolution parameter for the t-th feature map and the k-th feature map. Then the obtained feature map $\mathrm{F}_i^{o,k}$ is used by the decoded for the corresponding k-th task. By using the task-specific distillation feature map, the network can preserve more information for each individual task and is able to facilitate smooth convergence.

Multi-Task Distillation module C. The module C introduces attention mechanism for the distillation task. The attention mechanism [100] is successfully applied in various tasks such as image caption generation [169] and machine translation [97] for selecting useful information. Specifically, we utilize the attention mechanism for guiding the message passing between the feature maps of different tasks. Since the passed information flow is not always useful, the attention can act as a gate function to control the flow, in other words to make the network automatically learn to focus or to ignore information from other features. When we pass message to the k-th task, an attention map G_i^k is first produced from the corresponding set of feature maps F_i^k as follows:

$$\mathrm{G}_i^k \leftarrow \sigma(\mathrm{W}_g^k \otimes \mathrm{F}_i^k), \tag{5.2}$$

where W_g^k is the convolution parameter and σ is a sigmoid function for normalizing the

attention map. Then the message is passed with the attention map as follows:

$$\mathrm{F}_i^{o,k} \leftarrow \mathrm{F}_i^k + \sum_{t=1(\neq k)}^{T} \mathrm{G}_i^k \odot (\mathrm{W}_t \otimes \mathrm{F}_i^t), \tag{5.3}$$

where $\odot$ denotes element-wise multiplication.

5.3.5 Decoder Network Structure

For the final decoders, we use two deconvolutional layers to up-sample the distillation feature maps for pixel-level estimation. Since the distillation feature maps has a resolution of 1/4 as the input RGB image, each deconvolutional layer has 2 times upscaling in resolution and has the number of output channels reduced by half. Finally we use convolution to generate the score maps for each final task.

5.3.6 PAD-Net Optimization

We described the architecture details of the proposed multi-task guided prediction-and-distillation network. Now we describe the optimization and inference schemes of the overall network.

End-to-end network optimization. We have four intermediate prediction tasks, *i.e.* two discrete classification problems (scene parsing and contour prediction) and two continuous regression problems (surface normal estimation and depth estimation). However, we only require the annotations of the semantic labels and the depth, since the contour labels can be generated from the semantic labels and the surface normal can be calculated from the depth map. As our final target is to simultaneously perform the depth estimation and scene parsing, the whole network needs to optimize 6 losses with 4 different types. Specifically, we use a cross-entropy loss for the contour prediction task, a softmax loss for the scene parsing task and an Euclidean loss for the depth and surface normal estimation tasks. Since the groundtruth depth maps have invalid points, we masked the points during training. As in previous works [129, 142], we jointly learn the whole network with a linearly combined optimization objective $L_{all} = \sum_{i=1}^{6} w_i * L_i$, where L_i is the loss for the i-th task and w_i is the corresponding loss weight.

Inference. During the inference, We obtain the prediction results from the separate decoders. One important advantage of the PAD-Net is that it is able to incorporate rich domain knowledge from different predictions including scene semantic, depth, surface normal and object contours, while it only requires a single RGB image for the inference.

5.4 Experiments

To demonstrate the effectiveness of the proposed approach for simultaneous depth recovery and scene parsing, we conduct experiments on two publicly available benchmark datasets which provide both the depth and the semantic labels, including an indoor dataset NYU depth V2 (NYUD V2) [132] and an outdoor dataset Cityscapes [23]. In the following we describe the details of our experimental evaluation.

5.4.1 Experimental Setup

Datasets and Data Augmentation. The **NYUD V2** dataset [132] is a popular indoor RGBD dataset, which has been widely used for depth estimation [30] and semantic segmentation [45]. It contains 1449 pairs of RGB and depth images captured from a Kinect sensor, in which 795 pairs are used for training and the rest 654 for testing. Following [45], The training images are cropped to have a resolution of 560×425. The training data are augmented on the fly during the training phase. The RGB and depth images are scaled

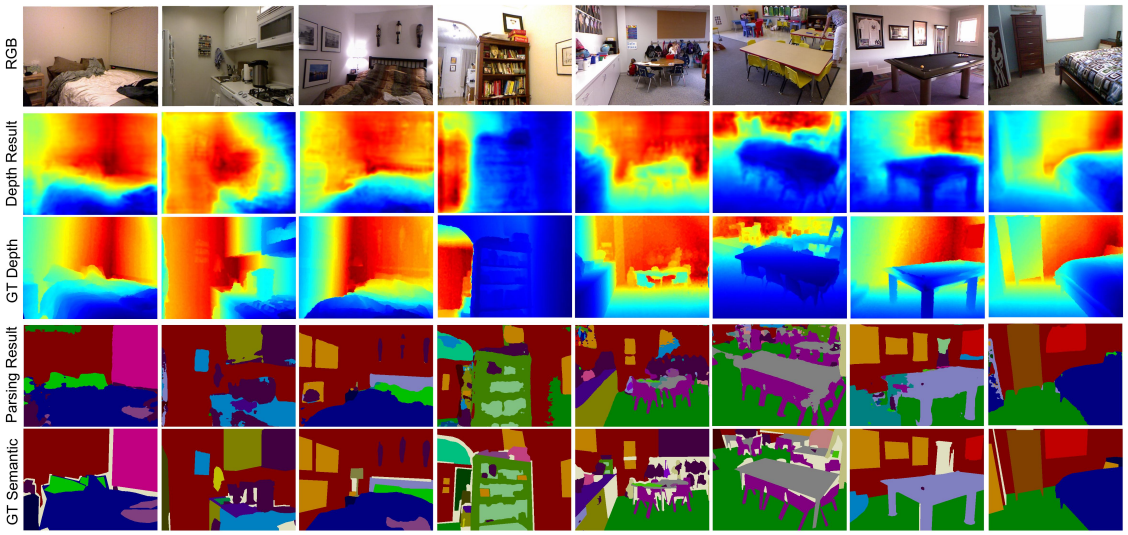


Figure 5.4: Quanlitative examples of depth prediction and 40-classes scene parsing results on the NYUD V2 dataset. The second and the four row are the estimated depth maps and the scene parsing results from the proposed PAD-Net, respectively.

with a randomly selected ratio in $\{1, 1.2, 1.5\}$ and the depth values are divided by the ratio. We also flip the training samples with a possibility of 0.5.

The **Cityscapes** [23] is a large-scale dataset mainly used for semantic urban scene understanding. The dataset is collected over 50 different cities spanning several months, and overall 19 semantic classes are annotated. The fine-annotated part consists of training, validation and test sets containing 2975, 500, and 1525 images, respectively. The dataset also provides pre-computed disparity depth maps associated with the rgb images. Similar to NYUD V2, we perform the data augmentation on the fly by scaling the images with a selected ratio in $\{0.5, 0.75, 1, 1.25, 1.75\}$ and randomly flipping them with a possibility of 0.5. As the images of the dataset have a high resolution (2048×1024), we crop the image with size of 640 for training due to the limitation of the GPU memory.

Evaluation Metrics. For evaluating the performance of the depth estimation, we use several quantitative metrics following previous works [30, 89, 164], including (a) mean relative error (rel): $\frac{1}{N}\sum_p \frac{|d_p - d_p^*|}{d_p}$; (b) root mean squared error (rms): $\sqrt{\frac{1}{N}\sum_p (d_p - d_p^*)^2}$; (c) mean log10 error (log10): $\frac{1}{N}\sum_i \|\log_{10}(d_p) - \log_{10}(d_p^*)\|$ and (d) accuracy with threshold t: percentage (%) of d_p^* subject to $\max(\frac{d_p^*}{d_p}, \frac{d_p}{d_p^*}) = \delta < t$ ($t \in [1.25, 1.25^2, 1.25^3]$), where d_p and d_p^* are the prediction and the groundtruth depth at the p-th pixel, respectively. For the evaluation of the semantic segmentation, we adopt three commonly used metrics, *i.e.* mean Intersection over Union (mIoU), mean accuracy and pixel accuracy. The mean IoU is calculated via averaging the Jaccard scores of all the predicted classes. The mean accuracy is the accuracy among all classes and pixel accuracy is the total accuracy of pixels regardless of the category. On the Cityscapes, both the pixel-level mIoU and instance-level mIoU are considered.

Implementation Details. The proposed network structure is implemented base on *Caffe* library [59] and on Nvidia Titan X GPUs. The front-end convolutional encoder of PAD-Net naturally supports any network structure. During the training, the front-end network is first initialized with parameters pre-trained with ImageNet for training, and the rest of the network is randomly intialized. The whole training process is performed

Table 5.1: Diagnostic experiments for the depth estimation task on NYUD V2 dataset. Distillation A, B, C represents the proposed three multi-task distillation modules.

Method	Error (lower is better)			Accuracy (higher is better)		
	rel	log10	rms	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Front-end + DE (baseline)	0.265	0.120	0.945	0.447	0.745	0.897
Front-end + DE + SP (baseline)	0.260	0.117	0.930	0.467	0.760	0.905
PAD-Net (Distillation A + DE)	0.248	0.112	0.892	0.513	0.798	0.921
PAD-Net (Distillation B + DE)	0.230	0.099	0.850	0.591	0.854	0.953
PAD-Net (Distillation C + DE)	0.221	0.094	0.813	0.619	0.882	0.965
PAD-Net (Distillation C + DE + SP)	**0.214**	**0.091**	**0.792**	**0.643**	**0.902**	**0.977**

Table 5.2: Diagnostic experiments for the scene parsing task on the NYUD V2 dataset.

Method	Mean IoU	Mean Accuracy	Pixel Accuracy
Front-end + SP (baseline)	0.291	0.301	0.612
Front-end + SP + DE (baseline)	0.294	0.312	0.615
PAD-Net (Distillation A + SP)	0.308	0.365	0.628
PAD-Net (Distillation B + SP)	0.317	0.411	0.638
PAD-Net (Distillation C + SP)	0.325	0.432	0.645
PAD-Net (Distillation C + DE + SP)	**0.331**	**0.448**	**0.647**

with two phases. In the first phase, we only optimize the front-end network with the scene parsing task and use a learning rate 0.001. After that, the whole network is jointly trained with multi-task losses and a lower learning rate of 10e-5 is used for a smooth convergence. As the final tasks are depth estimation and scene parsing, we set the loss weight of the contour prediction and surface normal estimation as 0.8. In the multi-task prediction module, N is set to 512. Total 60 epochs are used for NYUD V2, and 40 epochs for Cityscapes. Due to the sparse groundtruth depth maps of the Cityscapes dataset, the invalid points are masked out in the backpropagation. The network is optimized using stachastic gradient descent with the weight decay and the momentum set to 0.0005 and 0.99, respectively.

Table 5.3: Quantitative comparison with state-of-the-art methods methods on the semantic segmentation task on the NYUD V2 dataset. The methods 'Gupta *et al.*' [45] and 'Arsalan *et al.*' [101] jointly learn two tasks.

Method	Input Data Type	Mean IoU	Mean Accuracy	Pixel Accuracy
Deng *et al.* [26]	RGB + Depth	-	0.315	0.638
FCN [94]	RGB	0.292	0.422	0.600
FCN-HHA [94]	RGB + Depth	0.340	0.461	0.654
Eigen and Fergus [29]	RGB	0.341	0.451	0.656
Context [86]	RGB	0.406	0.536	0.700
Kong *et al.* [69]	RGB	0.445	-	0.721
RefineNet-Res152 [85]	RGB	0.465	0.589	0.736
Gupta *et al.* [45]	RGB + Depth	0.286	-	0.603
Arsalan *et al.* [101]	RGB	0.392	0.523	0.686
PAD-Net-ResNet50 (Ours)	RGB	**0.502**	**0.623**	**0.752**

Table 5.4: Quantitative comparison with state-of-the-art methods on the depth estimation task on NYUD V2 dataset. The methods 'Joint HCRF' [153] and 'Jafari *et al.*' [58] simultaneously learn the two tasks.

Method	# of Training	Error (lower is better)			Accuracy (higher is better)		
		rel	log10	rms	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Saxena *et al.* [127]	795	0.349	-	1.214	0.447	0.745	0.897
Karsch *et al.* [64]	795	0.35	0.131	1.20	-	-	-
Liu *et al.* [92]	795	0.335	0.127	1.06	-	-	-
Ladicky *et al.* [75]	795	-	-	-	0.542	0.829	0.941
Zhuo *et al.* [190]	795	0.305	0.122	1.04	0.525	0.838	0.962
Liu *et al.* [89]	795	0.230	0.095	0.824	0.614	0.883	0.975
Eigen *et al.* [30]	120K	0.215	-	0.907	0.611	0.887	0.971
Roi *et al.* [119]	795	0.187	0.078	0.744	-	-	-
Eigen and Fergus [29]	795	0.158	-	0.641	0.769	0.950	0.988
Laina *et al.* [77]	96K	0.129	0.056	0.583	0.801	0.950	0.986
Li *et al.*[81]	96K	0.139	0.058	0.505	0.820	0.960	0.989
Xu *et al.*[164]	4.7K	0.139	0.063	0.609	0.793	0.948	0.984
Xu *et al.*[164]	95K	0.121	**0.052**	0.586	0.811	0.950	0.986
Joint HCRF [153]	795	0.220	0.094	0.745	0.605	0.890	0.970
Jafari *et al.* [58]	795	0.157	0.068	0.673	0.762	0.948	0.988
PAD-Net-ResNet50 (Ours)	795	**0.120**	0.055	**0.582**	**0.817**	**0.954**	**0.987**

Table 5.5: Quantitative comparison results with the state-of-the-art methods on the Cityscapes *test* set. Our model is trained only on the fine-annotation dataset.

Method	IoU cla.	iIoU cla.	IoU cat.	iIoU cat.
SegNet [5]	0.561	0.342	0.798	0.664
CRF-RNN [188]	0.625	0.344	0.827	0.660
SiCNN [70]	0.663	0.449	0.850	0.712
DPN [93]	0.668	0.391	0.860	0.691
Dilation10 [177]	0.671	0.420	0.865	0.711
LRR [39]	0.697	0.480	0.882	0.747
DeepLab [19]	0.704	0.426	0.864	0.677
Piecewise [86]	0.716	0.517	0.873	0.741
PSPNet [186]	0.784	0.567	0.906	0.786
PAD-Net-ResNet101 (Ours)	0.803	0.588	0.908	0.785

5.4.2 Diagnostics Experiments

To deeply analyze the proposed approach and demonstrate its effectiveness, we conduct diagnostics experiments on both NYUD V2 and Cityscapes datasets. For the front-end network, according to the complexity of the dataset, we choose AlexNet [71] and ResNet-50 [50] network structures for NYUD V2 and Cityscapes, respectively.

Baseline methods and different variants of PAD-Net. To conduct the diagnostic experiments, we consider two baseline methods and different variants of the proposed PAD-Net. The baseline methods include: (i) Front-end + DE: performing the depth estimation (DE) task with the front-end CNN; (ii) Front-end + SP + DE: performing the scene parsing (SP) and the depth estimation tasks simultaneously with the front-end CNN. The different variants include: (i) PAD-Net (Distillation A + DE): PAD-Net performing the DE task using the distillation module A; (ii) PAD-Net (Distillation B + DE): similar to (i) while using the distillation module B; (iii) PAD-Net (Distillation B + DE): similar to (i) while using the distillation module C; (iv) PAD-Net (Distillation C + DE + SP):

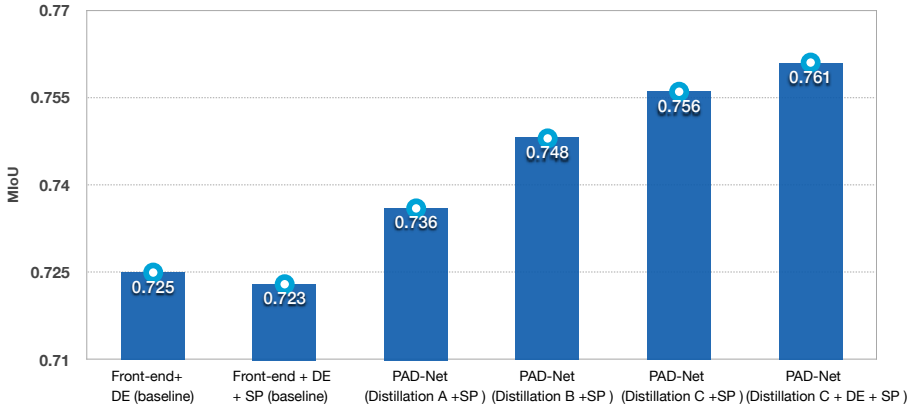


Figure 5.5: Diagnostic experiments of the proposed approach for the semantic segmentation task on the Cityscapes *val* dataset with ResNet-50 as the front-end backbone CNN.

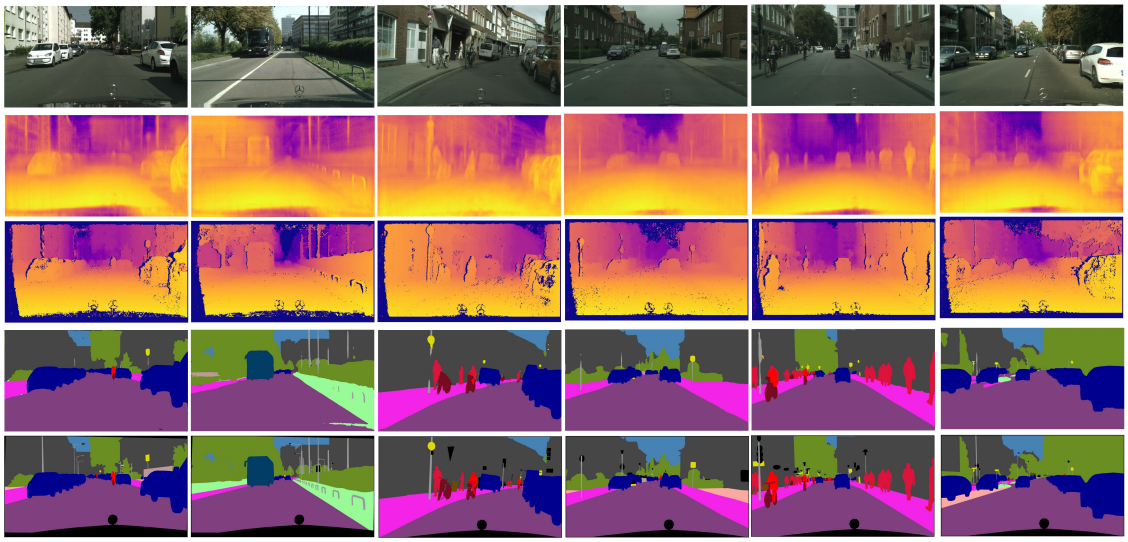


Figure 5.6: Quanlitative examples of depth prediction and 19-classes scene parsing results the Cityscapes dataset. The second and the fourth row correspond to the sparse depth and the semantic groundtruth, respectively.

performing DE and SP tasks simultaneously with the distillation module C.

Effect of direct multi-task learning. To investigate the effect of simultaneously optimizing two different task as previous works [101, 153], *i.e.* predicting two different tasks directly from the last scale feature map of the front-end CNN. We carry out experiments on both the NYUD V2 and Cityscapes datasets, as shown in Table. 5.1, 5.2 and Figure 5.5, respectively. It can be observed that on NYUD V2, the Front-end + DE + SP slightly outperforms the Front-end + DE, while on Cityscapes, the performance of Front-end + DE + SP is even decreased, which means that using a direct multi-task learning as traditional is probably not an effective means to facilitate each other the performance of different tasks.

Effect of multi-task distillation. We further evaluate the effect of the proposed three different distillation modules for incorporating information from different prediction tasks. Table 5.1 shows the results on the depth prediction task using PAD-Net embedded with the distillation module A, B and C. It can be seen that these three variants of PAD-Net are all obviously better than the two baseline methods, and the best one of ours,

PAD-Net (Distillation C + DE) is 4.4 and 2.3 points better than the baseline Front-end + DE + SP on the *rel* and on the *log10* metric respectively, and on the segmentation task on the same dataset, it is 3.1 points higher than the same baseline on the mIoU metric, which clearly demonstrates the effectiveness of the proposed multi-task distillation strategy. Similar performance gaps can be also observed on the segmentation task on Cityscapes in Figure 5.5. For comparing the different distillation modules, the message passing between different tasks (the module B and C) significantly boosts the the performance over the naive combination method (the module C). By using the attention guided scheme, the performance of the module C is further improved over the module B.

Effect of multi-task guided simultaneous prediction. We finally verify that the proposed multi-tasks guided prediction and distillation approach facilitates boosting the performance of both the depth estimation and scene parsing. The results of PAD-Net (Distillation C + DE + SP) clearly outperforms PAD-Net (Distillation C + DE) and PAD-Net (Distillation C + SP) in both the depth estimation task (Table 5.1) and the segmentation task (Tabel 5.2 and Figure 5.5). This shows that our design of PAD-Net can use multiple final tasks in learning more effective features. More importantly, PAD-Net (Distillation C + DE + SP) obtains remarkably better performance than the baseline Front-end + DE + SP, further demonstrating the superiority of the proposed PAD-Net compared with the methods directly using two tasks to learn a deep network.

5.4.3 State-of-the-art Comparison

Depth estimation. On the depth estimation task, we compare with several state-of-the-art methods, including: methods adopting hand-crafted features and deep representations [127, 127, 64, 75, 30, 29, 81, 77], and methods considering graphical modeling with CNN [92, 89, 190, 153, 164]. As shown in Table 5.4, PAD-Net using ResNet-50 network as the front-end achieves the best performance in all the measure metrics among all the comparison methods. It should be noted that our approach is trained only on the official training set with 795 images without using extra training data. More importantly, to compare with the methods working on joint learning the two tasks (Joint HCRF [153] and Jafari *et al.* [58]), our performance is remarkably higher than theirs, further verifying the advantage of the proposed approach. As the Cityscapes dataset only provides the disparity map, we do not quantitatively evaluate the depth estimation performance on this dataset. Figure 5.4 and 5.6 show qualitative examples of the depth estimation on the two datasets.

Scene parsing. For the scene parsing task, we quantitatively compare the performance with the state of the art methods both on NYUD V2 in Table 5.3 and on Cityscapes in Table 5.5. On NYUD V2, our PAD-Net-ResNet50 significantly outperforms the runner up competitor RefineNet-Res152 [85] with a 3.7 points gap on the mIoU metric. On the cityscapes, we train ours only on the fine-annotation training set, ours achieves a class-level mIoU of 0.803, which is 1.9 points better than the best competitor PSPNet trained on the same set. Qualitative scene parsing examples are shown in Figure 5.4 and 5.6.

5.5 Conclusion

We have presented the proposed PAD-Net for simultaneous depth estimation and scene parsing. The PAD-Net introduces a novel deep multi-task learning means, which first predicts several intermediate auxiliary tasks and then employs the multi-task predictions as guidance to facilitate optimizing the final main tasks. Three different multi-task distillation modules are developed to utilize the multi-task predictions more effectively. Our extensive

experiments on NYUD V2 and Cityscapes datasets demonstrated its effectiveness. We also provided new state of the art results on both the depth estimation and scene parsing tasks on NYUD V2, and top performance on Cityscapes scene parsing task.

CHAPTER 6

Conclusion

In this thesis we have explored the multi-modal and structured representation learning for multiple visual image and video understanding tasks, namely sketch-based image retrieval, pedestrian detection, monocular depth estimation and scene parsing.

Dictionary-based cross-modal representation learning usually encounters the non-convex optimization problem, which causes inferior convergence and leads to unsatisfied performance of the learned representations. We have addressed the problem in Chapter 2 via embedding a cross-modal self-paced curriculum learning scheme into a coupled dictionary learning framework to learn the training samples from easy to hard. The proposed learning method facilitates the algorithm to converge to better local minimal and thus produces more robust features. We have widely demonstrated the effectiveness of the approach on four publicly available SBIR datasets, *i.e.* CUFS, Flickr15k, QueenMary SBIR and TU-Berlin Extension. The proposed cross-paced learning strategy is also potentially useful to the popular deep learning models, which also have the non-convex optimization issue.

Following the progress of the shallow cross-modal representation learning, we have further developed a deep cross-modal feature learning and transferring framework for pedestrian detection in Chapter 3. Specifically, we devised a region reconstruction network for reconstructing one modality to another. By so doing, we can obtain the the cross-modal representations from the middle hidden layers. Then the learned representations are further transferred to a detection network for the pedestrian detection task. The proposed approach uses an unsupervised setting during training and only requires data from one single modality during testing. Extensive experiments on KAIST and Caltech Pedestrian datasets demonstrated the effectiveness of the proposed approach and established new state-of-the-art results.

In the above-mentioned cross-modal representation learning approaches, only single-scale representation is considered. To explore better use of the multi-scale deep representations for boosting the performance, we have devised a multi-scale structured model based on continuous CRFs in Chapter 4. The structured model is implemented as sequential neural networks for end-to-end learning to refine and fuse multi-scale predictions derived from the front-end CNN. The proposed model is generic and can be used for various regression problems involving continuous variables. We applied the approach for the monocular depth

estimation task. Experiments on three datasets, *i.e.* NYUD-V2, Make3D and KITTI show that the proposed approach significantly outperforms the baselines, and achieves state-of-the-art results.

Based on the encouraging progress on the cross-modal and structured feature learning for single tasks, we further investigated a more challenging problem, *i.e.* how to learn multi-modal predictions from one single modality and distill information from them to facilitate simultaneous optimization of multiple tasks. We have devised a joint deep network for this problem in Chapter 5. The network accepts RGB data as input, and intermediately produces different levels of predictions, such as object boundaries, semantic labels, surface normals and depth maps. These predictions are used as multi-modal input for the final optimization of the depth estimation and the scene parsing tasks. We obtained superior performance on NYUD-V2 and Cityscapes datasets. More importantly, the results showed that the proposed approach can effectively help producing consistent performance gain on both tasks when they are jointly optimized.

In summary, in this thesis we have studied different representation learning techniques for multiple visual understanding tasks. Our work suggests that the cross-paced feature learning, deep cross-modal representation learning and transferring, multi-scale deep structured learning and the multi-modal prediction and distillation indeed clearly improve the understanding of the visual data. In the future we will continue the research along the direction from the following possible aspects:

- The annotation of a large amount of visual data requires remarkable human efforts. To overcome this limitation, it is very promising to develop unsupervised or weakly supervised cross-modal representation learning frameworks, *e.g.* based on popular generative adversary networks.

- We explored the multi-scale structured learning on feature level or on prediction level. To obtain better structured output, we can jointly model and learn on both levels in a single network. Another issue in structured deep learning is the computational overhead. Thus it is also very important to study efficient schemes to speed up both the forward and the backward computation.

- For the multi-modal distillation, we can consider building a graphical model on the predictions of different modalities which involve both discrete and continuous variables, to better incorporate knowledge from the different modalities to aid the final optimization tasks.

Acknowledgements

I want to thank many important people who support and help me during my Ph.D. study. The thesis cannot be finished without them as my powerful backing. My supervisor Prof. Nicu Sebe, is a perfect mentor I have ever met in my academic experience. He does not perform only as a teacher, but also as a close friend to us. He always cares about our research and our life, and tries his best to help us when we meet any difficulties. Prof. Xiaogang Wang, was my supervisor when I was doing a visiting Ph.D. student at the Chinese University of Hong Kong. I am always impressed by the means that he thinks on the researches. He taught me how to discover valuable problems and how to conduct a deep research on them. Prof. Wanli Ouyang, was my co-supervisor when I was at CUHK. We were in the same office room, and worked very closely on the research projects. He is dedicated in the research and every time after the discussion with him, I would have deeper understanding of the problems. Prof. Elisa Ricci, my co-supervisor at the University of Trento, is another good friend during my research life in Trento. She helps me a lot in my research activities. She always gives me very useful advice especially when I get stuck in the research. Dr. Xavier Alameda-Pineda, is a very close collaborator during my whole Ph.D. study. He is a master of math. I am very pleased to have worked with him during these years and I learned a lot from him on how to mathematically treat a research problem and how it could be formalized. All the beloved mhuggers really bring me a lot of happiness. They are so friendly and helpful that I always forget that I am far away from my own hometown. I truly appreciate all the people mentioned above and wish everyone all the best. Finally, I would like to thank my parents and my wife for their support without reservation during my academic pursuit. I wish them all the happiness and good health.

Bibliography

[1] Adams, A., Baek, J., and Davis, M. A. (2010). Fast high-dimensional filtering using the permutohedral lattice. In *Computer Graphics Forum*, volume 29, pages 753–762.

[2] Alexe, B., Deselaers, T., and Ferrari, V. (2012). Measuring the objectness of image windows. *TPAMI*, 34(11):2189–2202.

[3] Angelova, A., Krizhevsky, A., Vanhoucke, V., Ogale, A., and Ferguson, D. (2015). Real-time pedestrian detection with deep network cascades. In *BMVC*.

[4] Arnab, A., Jayasumana, S., Zheng, S., and Torr, P. H. (2016). Higher order conditional random fields in deep neural networks. In *ECCV*. Springer.

[5] Badrinarayanan, V., Handa, A., and Cipolla, R. (2015). Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling. *arXiv preprint arXiv:1505.07293*.

[6] Beck, A. and Teboulle, M. (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imag. Sciences*, 2(1):183–202.

[7] Benenson, R., Mathias, M., Timofte, R., and Van Gool, L. (2012). Pedestrian detection at 100 frames per second. In *CVPR*.

[8] Benenson, R., Omran, M., Hosang, J., and Schiele, B. (2014). Ten years of pedestrian detection, what have we learned? In *ECCVW*.

[9] Bengio, Y., Louradour, J., Collobert, R., and Weston, J. (2009). Curriculum learning. In *ICML*.

[10] Bertasius, G., Shi, J., and Torresani, L. (2015). Deepedge: A multi-scale bifurcated deep network for top-down contour detection. In *CVPR*.

[11] Bimbo, A. D. and Pala, P. (1997). Visual image retrieval by elastic matching of user sketches. *TPAMI*, 19(2):121–132.

[12] Buyssens, P., Elmoataz, A., and Lézoray, O. (2012). Multiscale convolutional neural networks for vision–based classification of cells. In *ACCV*.

[13] Cai, Z., Fan, Q., Feris, R. S., and Vasconcelos, N. (2016). A unified multi-scale deep convolutional neural network for fast object detection. In *ECCV*.

[14] Cai, Z., Saberian, M., and Vasconcelos, N. (2015). Learning complexity-aware cascades for deep pedestrian detection. In *ICCV*.

[15] Cao, X., Wang, Z., Yan, P., and Li, X. (2013a). Transfer learning for pedestrian detection. *Neurocomputing*, 100:51–57.

[16] Cao, X., Zhang, H., Liu, S., Guo, X., and Lin, L. (2013b). Sym-fish: A symmetry-aware flip invariant sketch histogram shape descriptor. In *ICCV*.

[17] Chalechale, A., Naghdy, G., and Mertins, A. (2005). Sketch-based image matching using angular partitioning. *TSMC*, 35(1):28–41.

[18] Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2015). Semantic image segmentation with deep convolutional nets and fully connected crfs.

ICLR.

[19] Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2016a). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv preprint arXiv:1606.00915*.

[20] Chen, L.-C., Yang, Y., Wang, J., Xu, W., and Yuille, A. L. (2016b). Attention to scale: Scale-aware semantic image segmentation. *CVPR*.

[21] Chen, L.-C., Yang, Y., Wang, J., Xu, W., and Yuille, A. L. (2016c). Attention to scale: Scale-aware semantic image segmentation. In *CVPR*.

[22] Christoudias, C. M., Urtasun, R., Salzmann, M., and Darrell, T. (2010). Learning to recognize objects from unseen modalities. In *ECCV*.

[23] Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. (2016). The cityscapes dataset for semantic urban scene understanding. In *CVPR*.

[24] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *CVPR*.

[25] Delage, E., Lee, H., and Ng, A. Y. (2006). A dynamic bayesian network model for autonomous 3d reconstruction from a single indoor image. In *CVPR*.

[26] Deng, Z., Todorovic, S., and Jan Latecki, L. (2015). Semantic segmentation of rgbd images with mutex constraints. In *ICCV*.

[27] Dollár, P., Appel, R., Belongie, S., and Perona, P. (2014). Fast feature pyramids for object detection. *TPAMI*, 36(8):1532–1545.

[28] Dollár, P., Wojek, C., Schiele, B., and Perona, P. (2009). Pedestrian detection: A benchmark. In *CVPR*.

[29] Eigen, D. and Fergus, R. (2015). Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*.

[30] Eigen, D., Puhrsch, C., and Fergus, R. (2014). Depth map prediction from a single image using a multi-scale deep network. In *NIPS*.

[31] Eitz, M., Hays, J., and Alexa, M. (2012a). How do humans sketch objects? *TOG*, 31(4):44–1.

[32] Eitz, M., Hays, J., and Alexa, M. (2012b). How do humans sketch objects? *TOG*, 31(4):44.

[33] Eitz, M., Hildebrand, K., Boubekeur, T., and Alexa, M. (2010). An evaluation of descriptors for large-scale image retrieval from sketched feature lines. *Computers & Graphics*, 34(5):482–498.

[34] Eitz, M., Hildebrand, K., Boubekeur, T., and Alexa, M. (2011). Sketch-based image retrieval: Benchmark and bag-of-features descriptors. *TVCG*, 17(11):1624–1636.

[35] Everingham, M., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. (2010). The pascal visual object classes (VOC) challenge. *IJCV*, 88(2):303–338.

[36] Feng, F., Wang, X., and Li, R. (2014). Cross-modal retrieval with correspondence autoencoder. In *ACM MM*.

[37] Garg, R., Carneiro, G., and Reid, I. (2016). Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *ECCV*.

[38] Geiger, A., Lenz, P., Stiller, C., and Urtasun, R. (2013). Vision meets robotics: The kitti dataset. *IJRR*.

[39] Ghiasi, G. and Fowlkes, C. C. (2016). Laplacian pyramid reconstruction and refinement for semantic segmentation. In *ECCV*. Springer.

[40] Girshick, R. (2015). Fast r-cnn. In *ICCV*.

[41] Godard, C., Mac Aodha, O., and Brostow, G. J. (2017). Unsupervised monocular depth estimation with left-right consistency. In *CVPR*.

[42] González, A., Fang, Z., Socarras, Y., Serrat, J., Vázquez, D., Xu, J., and López, A. M. (2016). Pedestrian detection at day/night time with visible and fir cameras: A comparison. *Sensors*, 16(6):820.

[43] Guo, J., Wang, C., and Chao, H. (2015). Building effective representations for sketch recognition. In *AAAI*.

[44] Gupta, S., Arbelaez, P., and Malik, J. (2013). Perceptual organization and recognition of indoor scenes from rgb-d images. In *CVPR*.

[45] Gupta, S., Girshick, R., Arbeláez, P., and Malik, J. (2014). Learning rich features from rgb-d images for object detection and segmentation. In *ECCV*.

[46] Gupta, S., Hoffman, J., and Malik, J. (2016). Cross modal distillation for supervision transfer. In *CVPR*.

[47] Han, H., Jain, A. K., Shan, S., and Chen, X. (2017). Heterogeneous face attribute estimation: A deep multi-task learning approach. *arXiv preprint arXiv:1706.00906*.

[48] Hariharan, B., Arbeláez, P., Girshick, R., and Malik, J. (2014). Simultaneous detection and segmentation. In *ECCV*, pages 297–312.

[49] Hariharan, B., Arbeláez, P., Girshick, R., and Malik, J. (2015). Hypercolumns for object segmentation and fine-grained localization. In *CVPR*.

[50] He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*.

[51] Hoffman, J., Gupta, S., and Darrell, T. (2016). Learning with side information through modality hallucination. In *CVPR*.

[52] Hoiem, D., Efros, A. A., and Hebert, M. (2005a). Automatic photo pop-up. *ACM TOG*, 24(3):577–584.

[53] Hoiem, D., Efros, A. A., and Hebert, M. (2005b). Geometric context from a single image. In *ICCV*.

[54] Hosang, J., Omran, M., Benenson, R., and Schiele, B. (2015). Taking a deeper look at pedestrians. In *CVPR*.

[55] Hu, R. and Collomosse, J. (2013). A performance evaluation of gradient field hog descriptor for sketch based image retrieval. *CVIU*, 117(7):790–806.

[56] Huang, D.-A. and Wang, Y.-C. F. (2013). Coupled dictionary and feature space learning with applications to cross-domain image synthesis and recognition. In *ICCV*.

[57] Hwang, S., Park, J., Kim, N., Choi, Y., and So Kweon, I. (2015). Multispectral pedestrian detection: Benchmark dataset and baseline. In *CVPR*.

[58] Jafari, O. H., Groth, O., Kirillov, A., Yang, M. Y., and Rother, C. (2017). Analyzing modular cnn architectures for joint depth prediction and semantic segmentation. *arXiv preprint arXiv:1702.08009*.

[59] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*.

[60] Jiang, L., Meng, D., Mitamura, T., and Hauptmann, A. G. (2014a). Easy samples first: Self-paced reranking for zero-example multimedia search. In *ACM MM*.

[61] Jiang, L., Meng, D., Yu, S.-I., Lan, Z., Shan, S., and Hauptmann, A. (2014b). Self-paced learning with diversity. In *NIPS*.

[62] Jiang, L., Meng, D., Zhao, Q., Shan, S., and Hauptmann, A. G. (2015). Self-paced curriculum learning. In *AAAI*.

[63] Karpathy, A., Joulin, A., and Li, F. F. (2014). Deep fragment embeddings for bidirectional image sentence mapping. In *NIPS*.

[64] Karsch, K., Liu, C., and Kang, S. B. (2014). Depth transfer: Depth extraction from video using non-parametric sampling. *IEEE TPAMI*, 36(11):2144–2158.

[65] Kato, T., Kurita, T., Otsu, N., and Hirata, K. (1992). A sketch retrieval method for full color image database-query by visual example. In *ICPR*.

[66] Kendall, A., Gal, Y., and Cipolla, R. (2017). Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. *arXiv preprint arXiv:1705.07115*.

[67] Knöbelreiter, P., Reinbacher, C., Shekhovtsov, A., and Pock, T. (2016). End-to-end training of hybrid cnn-crf models for stereo. *arXiv preprint arXiv:1611.10229*.

[68] Koltun, V. (2011). Efficient inference in fully connected crfs with gaussian edge potentials. *NIPS*.

[69] Kong, S. and Fowlkes, C. (2017). Recurrent scene parsing with perspective understanding in the loop. *arXiv preprint arXiv:1705.07238*.

[70] Krešo, I., Čaušević, D., Krapac, J., and Šegvić, S. (2016). Convolutional scale invariance for semantic segmentation. In *GCPR*. Springer.

[71] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *NIPS*.

[72] Kuga, R., Kanezaki, A., Samejima, M., Sugano, Y., and Matsushita, Y. (2017). Multi-task learning using multi-modal encoder-decoder networks with shared skip connections. In *ICCVW*.

[73] Kumar, M. P., Packer, B., and Koller, D. (2010). Self-paced learning for latent variable models. In *NIPS*.

[74] Kuznietsov, Y., Stückler, J., and Leibe, B. (2017). Semi-supervised deep learning for monocular depth map prediction. In *CVPR*.

[75] Ladicky, L., Shi, J., and Pollefeys, M. (2014). Pulling things out of perspective. In *CVPR*.

[76] Lafferty, J., McCallum, A., Pereira, F., et al. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*.

[77] Laina, I., Rupprecht, C., Belagiannis, V., Tombari, F., and Navab, N. (2016). Deeper depth prediction with fully convolutional residual networks. *arXiv preprint arXiv:1606.00373*.

[78] Lazebnik, S., Schmid, C., and Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*.

[79] Lee, H., Battle, A., Raina, R., and Ng, A. Y. (2006). Efficient sparse coding algorithms. In *NIPS*.

[80] Lee, Y. J. and Grauman, K. (2011). Learning the easy things first: Self-paced visual category discovery. In *CVPR*.

[81] Li, B., Dai, Y., and He, M. (2017). Monocular depth estimation with hierarchical fusion of dilated cnns and soft-weighted-sum inference. *arXiv preprint arXiv:1708.02287*.

[82] Li, B., Shen, C., Dai, Y., van den Hengel, A., and He, M. (2015a). Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. In *CVPR*.

[83] Li, J., Liang, X., Shen, S., Xu, T., and Yan, S. (2015b). Scale-aware fast r-cnn for

pedestrian detection. *arXiv preprint arXiv:1510.08160*.

[84] Li, Y., Hospedales, T. M., Song, Y.-Z., and Gong, S. (2014). Fine-grained sketch-based image retrieval by matching deformable part models. In *BMVC*.

[85] Lin, G., Milan, A., Shen, C., and Reid, I. (2016a). Refinenet: Multi-path refinement networks with identity mappings for high-resolution semantic segmentation. *arXiv preprint arXiv:1611.06612*.

[86] Lin, G., Shen, C., van den Hengel, A., and Reid, I. (2016b). Efficient piecewise training of deep structured models for semantic segmentation. In *CVPR*.

[87] Lin, Y.-L., Huang, C.-Y., Wang, H.-J., and Hsu, W.-C. (2013). 3d sub-query expansion for improving sketch-based multi-view image retrieval. In *ICCV*.

[88] Liu, B., Gould, S., and Koller, D. (2010). Single image depth estimation from predicted semantic labels. In *CVPR*.

[89] Liu, F., Shen, C., and Lin, G. (2015a). Deep convolutional neural fields for depth estimation from a single image. In *CVPR*.

[90] Liu, F., Shen, C., Lin, G., and Reid, I. (2016). Learning depth from single monocular images using deep convolutional neural fields. *IEEE TPAMI*, 38(10):2024–2039.

[91] Liu, L., Shen, F., Shen, Y., Liu, X., and Shao, L. (2017). Deep sketch hashing: Fast free-hand sketch-based image retrieval. In *CVPR*.

[92] Liu, M., Salzmann, M., and He, X. (2014). Discrete-continuous depth estimation from a single image. In *CVPR*.

[93] Liu, Z., Li, X., Luo, P., Loy, C.-C., and Tang, X. (2015b). Semantic image segmentation via deep parsing network. In *ICCV*.

[94] Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *CVPR*.

[95] Long, M. and Wang, J. (2015). Learning transferable features with deep adaptation networks. *ICML*.

[96] Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *ICCV*.

[97] Luong, M.-T., Pham, H., and Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

[98] Mairal, J., Bach, F., Ponce, J., Sapiro, G., and Zisserman, A. (2009). Non-local sparse models for image restoration. In *ICCV*.

[99] Misra, I., Shrivastava, A., Gupta, A., and Hebert, M. (2016). Cross-stitch networks for multi-task learning. In *CVPR*.

[100] Mnih, V., Heess, N., Graves, A., et al. (2014). Recurrent models of visual attention. In *NIPS*.

[101] Mousavian, A., Pirsiavash, H., and Košecká, J. (2016). Joint semantic segmentation and depth estimation with deep convolutional networks. In *3DV*.

[102] Nam, W., Dollár, P., and Han, J. H. (2014). Local decorrelation for improved pedestrian detection. In *NIPS*.

[103] Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., and Ng, A. Y. (2011). Multimodal deep learning. In *ICML*.

[104] Noh, H., Hong, S., and Han, B. (2015). Learning deconvolution network for semantic segmentation. In *ICCV*.

[105] Ouyang, W. and Wang, X. (2012). A discriminative deep model for pedestrian detection with occlusion handling. In *CVPR*.

[106] Ouyang, W. and Wang, X. (2013). Single-pedestrian detection aided by multi-pedestrian detection. In *CVPR*.

[107] Ouyang, W., Zeng, X., and Wang, X. (2013). Modeling mutual visibility relationship in pedestrian detection. In *CVPR*.

[108] Paisitkriangkrai, S., Shen, C., and van den Hengel, A. (2014). Strengthening the effectiveness of pedestrian detection with spatially pooled features. In *ECCV*.

[109] Paisitkriangkrai, S., Shen, C., and van den Hengel, A. (2015). Pedestrian detection with spatially pooled features and structured ensemble learning. *TPAMI*, PP(99):1–1.

[110] Papandreou, G., Chen, L.-C., Murphy, K., and Yuille, A. L. (2015). Weakly-and semi-supervised learning of a dcnn for semantic image segmentation. *arXiv preprint arXiv:1502.02734*.

[111] Pentina, A., Sharmanska, V., and Lampert, C. H. (2015). Curriculum learning of multiple tasks. In *CVPR*.

[112] Porzi, L., Buló, S. R., Penate-Sanchez, A., Ricci, E., and Moreno-Noguer, F. (2017). Learning depth-aware deep representations for robotic perception. *IEEE Robotics and Automation Letters*, 2(2):468–475.

[113] Premebida, C., Carreira, J., Batista, J., and Nunes, U. (2014). Pedestrian detection combining rgb and dense lidar data. In *IROS*.

[114] Qi, Y., Song, Y.-Z., Xiang, T., Zhang, H., Hospedales, T., Li, Y., and Guo, J. (2015). Making better use of edges via perceptual grouping. In *CVPR*.

[115] Qi, Y., Song, Y.-Z., Zhang, H., and Liu, J. (2016). Sketch-based image retrieval via siamese convolutional neural network. In *ICIP*.

[116] Ranjan, R., Patel, V. M., and Chellappa, R. (2016). Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *arXiv preprint arXiv:1603.01249*.

[117] Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*.

[118] Ristovski, K., Radosavljevic, V., Vucetic, S., and Obradovic, Z. (2013). Continuous conditional random fields for efficient regression in large fully connected graphs. In *AAAI*.

[119] Roy, A. and Todorovic, S. (2016). Monocular depth estimation using neural regression forest. In *CVPR*.

[120] Ruder, S., Bingel, J., Augenstein, I., and Søgaard, A. (2017). Sluice networks: Learning what to share between loosely related tasks. *arXiv preprint arXiv:1705.08142*.

[121] Saavedra, J. M. (2014). Sketch based image retrieval using a soft computation of the histogram of edge local orientations (s-helo). In *ICIP*.

[122] Saavedra, J. M., Barrios, J. M., and Orand, S. (2015). Sketch based image retrieval using learned keyshapes (LKS). In *BMVC*.

[123] Saavedra, J. M. and Bustos, B. (2010). An improved histogram of edge local orientations for sketch-based image retrieval. In *Pattern Recognition*, pages 432–441. Springer.

[124] Sangkloy, P., Burnell, N., Ham, C., and Hays, J. (2016). The sketchy database: learning to retrieve badly drawn bunnies. *TOG*, 35(4):119.

[125] Saxena, A., Chung, S. H., and Ng, A. Y. (2005). Learning depth from single monocular images. In *NIPS*.

[126] Saxena, A., Chung, S. H., and Ng, A. Y. (2008). 3-d depth reconstruction from a

single still image. *IJCV*, 76(1):53–69.

[127] Saxena, A., Sun, M., and Ng, A. Y. (2009). Make3d: Learning 3d scene structure from a single still image. *IEEE TPAMI*, 31(5):824–840.

[128] Schwing, A. G. and Urtasun, R. (2015). Fully connected deep structured networks. *arXiv preprint arXiv:1503.02351*.

[129] Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., and LeCun, Y. (2013a). Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*.

[130] Sermanet, P., Kavukcuoglu, K., Chintala, S., and LeCun, Y. (2013b). Pedestrian detection with unsupervised multi-stage feature learning. In *CVPR*.

[131] Sharma, A. and Jacobs, D. W. (2011). Bypassing synthesis: Pls for face recognition with pose, low-resolution and sketch. In *CVPR*.

[132] Silberman, N., Hoiem, D., Kohli, P., and Fergus, R. (2012). Indoor segmentation and support inference from rgbd images. In *ECCV*.

[133] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[134] Socarrás, Y., Ramos, S., Vázquez, D., López, A. M., and Gevers, T. (2011). Adapting pedestrian detection from synthetic to far infrared images. In *ICCVW*.

[135] Socher, R., Ganjoo, M., Manning, C. D., and Ng, A. (2013). Zero-shot learning through cross-modal transfer. In *NIPS*.

[136] Sohn, K., Lee, H., and Yan, X. (2015). Learning structured output representation using deep conditional generative models. In *NIPS*.

[137] Srivastava, N. and Salakhutdinov, R. R. (2012). Multimodal learning with deep boltzmann machines. In *NIPS*.

[138] Sun, X., Wang, C., Xu, C., and Zhang, L. (2013). Indexing billions of images for sketch-based retrieval. In *ACM MM*.

[139] Supancic, J. S. and Ramanan, D. (2013). Self-paced learning for long-term tracking. In *CVPR*.

[140] Tang, X. and Wang, X. (2004). Face sketch recognition. *TCSVT*, 14(1):50–57.

[141] Tang, Y., Yang, Y.-B., and Gao, Y. (2012). Self-paced dictionary learning for image classification. In *ACM MM*.

[142] Teichmann, M., Weber, M., Zoellner, M., Cipolla, R., and Urtasun, R. (2016). Multinet: Real-time joint semantic reasoning for autonomous driving. *arXiv preprint arXiv:1612.07695*.

[143] Tenenbaum, J. B. and Freeman, W. T. (2000). Separating style and content with bilinear models. *Neural Computation*, 12(6):1247–1283.

[144] Tian, Y., Luo, P., Wang, X., and Tang, X. (2015a). Deep learning strong parts for pedestrian detection. In *ICCV*.

[145] Tian, Y., Luo, P., Wang, X., and Tang, X. (2015b). Pedestrian detection aided by deep learning semantic tasks. In *CVPR*.

[146] Vanderbei, R. J. (1999). Loqo: An interior point code for quadratic programming. *Optimization methods and software*, 11(1-4):451–484.

[147] Viola, P. and Jones, M. J. (2004). Robust real-time face detection. *IJCV*, 57(2):137–154.

[148] Viola, P., Jones, M. J., and Snow, D. (2005). Detecting pedestrians using patterns of motion and appearance. *IJCV*, 63(2):153–161.

[149] Wang, B., Yang, Y., Xu, X., Hanjalic, A., and Shen, H. T. (2017). Adversarial cross-modal retrieval. In *ACM MM*, pages 154–162.

[150] Wang, F., Kang, L., and Li, Y. (2015a). Sketch-based 3d shape retrieval using convolutional neural networks. In *CVPR*.

[151] Wang, K., He, R., Wang, W., Wang, L., and Tan, T. (2013). Learning coupled feature spaces for cross-modal matching. In *ICCV*.

[152] Wang, M., Li, W., and Wang, X. (2012a). Transferring a generic pedestrian detector towards specific scenes. In *CVPR*.

[153] Wang, P., Shen, X., Lin, Z., Cohen, S., Price, B., and Yuille, A. (2015b). Towards unified depth and semantic prediction from a single image. In *CVPR*.

[154] Wang, S., Zhang, L., Liang, Y., and Pan, Q. (2012b). Semi-coupled dictionary learning with applications to image super-resolution and photo-sketch synthesis. In *CVPR*.

[155] Wang, X. and Tang, X. (2009). Face photo-sketch synthesis and recognition. *TPAMI*, 31(11):1955–1967.

[156] Xia, F., Wang, P., Chen, L.-C., and Yuille, A. L. (2016). Zoom better to see clearer: Human and object parsing with hierarchical auto-zoom net. In *ECCV*. Springer.

[157] Xiao, C., Wang, C., Zhang, L., and Zhang, L. (2015). Sketch-based image retrieval via shape words. In *ICMR*.

[158] Xie, S. and Tu, Z. (2015). Holistically-nested edge detection. In *ICCV*.

[159] Xu, C., Tao, D., and Xu, C. (2015a). Multi-view self-paced learning for clustering. In *IJCAI*.

[160] Xu, D., Alameda-Pineda, X., Song, J., Ricci, E., and Sebe, N. (2016a). Academic coupled dictionary learning for sketch-based image retrieval. In *ACM MM*.

[161] Xu, D., Ouyang, W., Alameda-Pineda, X., Ricci, E., Wang, X., and Sebe, N. (2017a). Learning deep structured multi-scale features using attention-gated crfs for contour prediction.

[162] Xu, D., Ouyang, W., Ricci, E., Wang, X., and Sebe, N. (2017b). Learning cross-modal deep representations for robust pedestrian detection. In *CVPR*.

[163] Xu, D., Ouyang, W., Wang, X., and Sebe, N. (2018a). Pad-net: Multi-tasks guided prediciton-and-distillation network for simultaneous depth estimation and scene parsing. In *CVPR*.

[164] Xu, D., Ricci, E., Ouyang, W., Wang, X., and Sebe, N. (2017c). Multi-scale continuous crfs as sequential deep networks for monocular depth estimation. In *CVPR*.

[165] Xu, D., Ricci, E., Yan, Y., Song, J., and Sebe, N. (2015b). Learning deep representations of appearance and motion for anomalous event detection. In *BMVC*.

[166] Xu, D., Song, J., Xavier, A.-P., Ricci, E., and Sebe, N. (2016b). Multi-paced dictionary learning for cross-domain retrieval and recognition. In *ICPR*.

[167] Xu, D., Wang, W., Tang, H., Sebe, N., and Ricci, E. (2018b). Structured attention guided convolutional neural fields for monocular depth estimation. In *CVPR*.

[168] Xu, D., Yan, Y., Ricci, E., and Sebe, N. (2017d). Detecting anomalous events in videos by learning deep representations of appearance and motion. *CVIU*, 156:117–127.

[169] Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., and Bengio, Y. (2015c). Show, attend and tell: Neural image caption generation with visual attention. In *ICML*.

[170] Xu, P., Yin, Q., Qi, Y., Song, Y.-Z., Ma, Z., Wang, L., and Guo, J. (2016c).

Instance-level coupled subspace learning for fine-grained sketch-based image retrieval. In *ECCV*.

[171] Yan, J., Zhang, X., Lei, Z., Liao, S., and Li, S. (2013). Robust multi-resolution pedestrian detection in traffic scenes. In *CVPR*.

[172] Yan, Y., Yang, Y., Shen, H., Meng, D., Liu, G., Hauptmann, A., and Sebe, N. (2015). Complex event detection via event oriented dictionary learning. In *AAAI*.

[173] Yang, B., Yan, J., Lei, Z., and Li, S. Z. (2015a). Convolutional channel features. In *ICCV*.

[174] Yang, J., Wright, J., Huang, T. S., and Ma, Y. (2010). Image super-resolution via sparse representation. *TIP*, 19(11):2861–2873.

[175] Yang, Y., Wang, Z., and Wu, F. (2015b). Exploring prior knowledge for pedestrian detection. In *BMVC*.

[176] Yao, J., Fidler, S., and Urtasun, R. (2012). Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation. In *CVPR*.

[177] Yu, F. and Koltun, V. (2015). Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*.

[178] Yu, Q., Liu, F., Song, Y.-Z., Xiang, T., Hospedales, T. M., and Loy, C.-C. (2016). Sketch me that shoe. In *CVPR*.

[179] Yu, Q., Yang, Y., Song, Y.-Z., Xiang, T., and Hospedales, T. (2015). Sketch-a-net that beats humans. In *BMVC*.

[180] Zeng, X., Ouyang, W., Wang, M., and Wang, X. (2014). Deep learning of scene-specific classifier for pedestrian detection. In *ECCV*.

[181] Zhai, X., Peng, Y., and Xiao, J. (2013). Heterogeneous metric learning with joint graph regularization for cross-media retrieval. In *AAAI*.

[182] Zhang, H., Liu, S., Zhang, C., Ren, W., Wang, R., and Cao, X. (2016a). Sketchnet: Sketch classification with web images. In *CVPR*.

[183] Zhang, L., Lin, L., Liang, X., and He, K. (2016b). Is faster r-cnn doing well for pedestrian detection? In *ECCV*.

[184] Zhang, S., Bauckhage, C., and Cremers, A. (2014). Informed haar-like features improve pedestrian detection. In *CVPR*.

[185] Zhang, S., Benenson, R., and Schiele, B. (2015). Filtered channel features for pedestrian detection. In *CVPR*.

[186] Zhao, H., Shi, J., Qi, X., Wang, X., and Jia, J. (2016). Pyramid scene parsing network. *arXiv preprint arXiv:1612.01105*.

[187] Zhao, Q., Meng, D., Jiang, L., Xie, Q., Xu, Z., and Hauptmann, A. G. (2015). Self-paced learning for matrix factorization. In *AAAI*.

[188] Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C., and Torr, P. H. (2015). Conditional random fields as recurrent neural networks. In *ICCV*.

[189] Zhou, T., Brown, M., Snavely, N., and Lowe, D. G. (2017). Unsupervised learning of depth and ego-motion from video. In *CVPR*.

[190] Zhuo, W., Salzmann, M., He, X., and Liu, M. (2015). Indoor scene structure analysis for single image depth estimation. In *CVPR*.