UNIVERSITY OF CALIFORNIA, MERCED

Learning Correspondence from Images, Videos and Texts

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Electrical Engineering and Computer Science

by

Taihong Xiao

Committee in charge:

Professor Ming-Hsuan Yang, Chair
Professor Shawn Newsam
Professor Sungjin Im
Dr Sifei Liu

2023

The dissertation of Taihong Xiao is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

(Professor Shawn Newsam)

(Professor Sungjin Im)

(Dr Sifei Liu)

(Professor Ming-Hsuan Yang, Chair)

University of California, Merced

2023

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

To begin with, I wish to convey my deepest gratitude to my advisor, Professor Ming-Hsuan Yang. My Ph.D. journey has immensely benefited from his unwavering support and invaluable guidance. He has patiently listened to my ideas, offered constructive feedback on my research projects, and diligently revised paper drafts throughout my doctoral years. Furthermore, he has inspired me to evolve into an integrated and independent researcher capable of identifying problems and potential challenges, thinking critically, cultivating collaborations with fellow researchers, and effectively managing both upward and downward. These aspects have proven pivotal not only to my Ph.D. studies but also to my long-term career.

I extend heartfelt thanks to my thesis committee members - Shawn Newsam, Sungjin Im, and Sifei Liu, for generously dedicating their time and providing kind advice and constructive feedback on my thesis. Their mentorship has enriched my knowledge and guided me in enhancing the quality of my research.

I am fortunate to have collaborated with numerous talented mentors, peers, and colleagues. I extend my appreciation to Yi-Hsuan Tsai, Liangliang Cao, Kihyuk Sohn, Jinwei Yuan, Deqing Sun, Sifei Liu, Zhiding Yu, Qifei Wang, Shalini De Mello, Xin-Yu Zhang, Zirui Wang, Jiahui Yu, Hsin-Ping Huang, Zhiwei Lin, and Tingting Liang. I am grateful to all of you for engaging in stimulating discussions, improving my paper writing, and enhancing the quality of my presentations.

Finally, I thank my family and friends for their unwavering love and support. Despite the physical distance that separates us, my family has always been there for me, offering their support and love whenever I feel excited or frustrated. I would also like to sincerely thank Erik Suh, who provided the warmest support and friendship to an international student living abroad.

VITA

2011-2015	B. S. in Mathematics, Shandong University
2015-2018	M. S. in Applied Mathematics, Peking University
2018-2023	Ph. D. in Electrical Engineer and Computer Science, University of California, Merced

PUBLICATIONS

Taihong Xiao, Yi-Hsuan Tsai, Kihyuk Sohn, Manmohan Chandraker, Ming-Hsuan Yang, “Adversarial Learning of Privacy-Preserving and Task-Oriented Representations”, *Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

Taihong Xiao, Jinwei Yuan, Deqing Sun, Qifei Wang, Xin-Yu Zhang, Kehan Xu, Ming-Hsuan Yang, “Learnable Cost Volume Using the Cayley Representation”, *European Conference on Computer Vision (ECCV)*, 2020

Taihong Xiao, Xin-Yu Zhang, Haolin Jia, Ming-Ming Cheng, Ming-Hsuan Yang, “Semi-Supervised Learning with Meta-Gradient”, *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2021

Xin-Yu Zhang, Kai Zhao, **Taihong Xiao**, Ming-Ming Cheng, Ming-Hsuan Yang, “Structured Sparsification with Joint Optimization of Group Convolution and Channel Shuffle”, *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2021

Taihong Xiao, Sifei Liu, Shalini De Mello, Zhiding Yu, Jan Kautz, Ming-Hsuan Yang, “Learning Contrastive Representation for Semantic Correspondence”, *International Journal of Computer Vision (IJCV)*, 2022

Hsin-Ping Huang, Deqing Sun, Yaojie Liu, Wen-Sheng Chu, **Taihong Xiao**, Jinwei Yuan, Hartwig Adam, Ming-Hsuan Yang, “Adaptive Transformers for Robust Few-shot Cross-domain Face Anti-spoofing”, *European Conference on Computer Vision (ECCV)*, 2022

Zhiwei Lin*, Tingting Liang*, **Taihong Xiao***, Yongtao Wang, Zhi Tang, Ming-Hsuan Yang, “FlowNAS: Neural Architecture Search for Optical Flow Estimation”, *International Journal of Computer Vision (IJCV)*, 2023

Taihong Xiao, Zirui Wang, Liangliang Cao, Jiahui Yu, Shengyang Dai, Ming-Hsuan Yang, “Exploiting Category Names for Few-Shot Classification with Vision-Language Models”, *International Conference on Learning Representations (ICLR), Multimodal Representation Learning Workshop*, 2023

ABSTRACT OF THE DISSERTATION

Learning Correspondence from Images, Videos and Texts

by

Taihong Xiao

Doctor of Philosophy in Electrical Engineering and Computer Science

University of California Merced, 2023

Professor Ming-Hsuan Yang, Chair

In computer vision, learning correspondence is a pivotal and fundamental challenge with far-reaching applications. Correspondence encapsulates a measure of similarity between disparate entities, spanning images, videos, and texts. As deep neural networks have demonstrated significant success in computer vision over the past few years, inferring correspondence has been posed as a representation learning task. We learn useful feature representations and infer correspondence with deep neural networks. In this thesis, we undertake the task of learning various types of correspondence and exploring their applications.

First, we consider acquiring dense low-level correspondence between successive video frames, where optical flow represents temporal pixel-level correspondences. Within the landscape of deep optical flow estimation methodologies, the cost volume emerges as a linchpin, encoding the vital pixel-level correlation information. Our contribution comes as a learnable cost volume (LCV) layer, leveraging a positive definite kernel matrix and optimizing its learning through Cayley representations. The proposed LCV is a lightweight module and can be easily plugged into existing models to replace the conventional cost volume. It reduces flow estimation errors and improves the model's robustness against illumination variations, noise, and adversarial input perturbations.

Second, we delve into semantic correspondence across distinct images, a task more challenging than optical flow estimation. Here, we confront the complexities stemming from vast variations in appearance, scale, and pose, even among objects in the same category. We introduce an affinity matrix to represent semantic similarity between images. Our novel approach harnesses multi-level contrastive learning for semantic matching. It leverages image-level contrastive learning to guide convolutional features in locating correspondence between similar objects. Further, we enhance performance through pixel-level cross-instance cycle consistency. This methodology outperforms prevailing approaches in this domain.

Finally, we explore the correspondence between images and text, crucial in vision-language foundation models bridging disparate modalities. These models employ visual and textual encoders, mapping both modalities into a shared embedding space. While pretrained representations from extensive data yield impressive zero-shot performance in tasks like image classification, their potential wanes when dealing with few examples per category. To address this challenge, we propose a category name initialization method that initializes the visual classification head with text embeddings of category names. Extensive experimental results show that the category name initialization method propels our model to achieve state-of-the-art results in various few-shot image classification benchmarks.

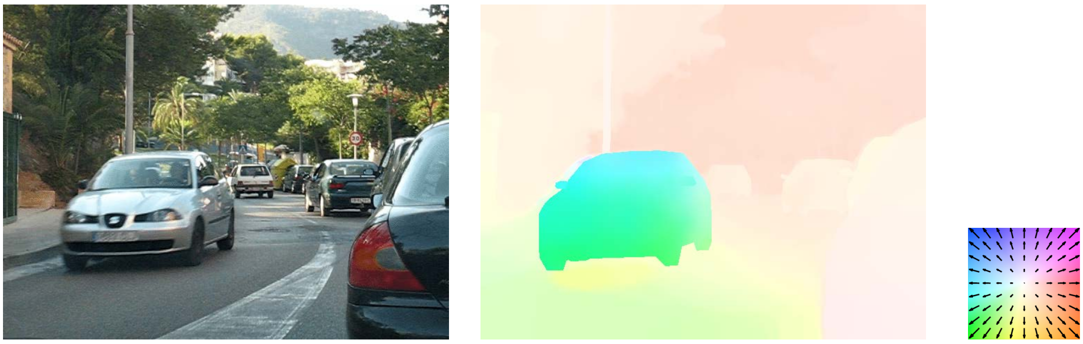
Chapter 1

Introduction

Learning a meaningful correspondence between two or more entities is a fundamental task. The problem can be generally stated as: given two entities, find a meaningful relation or mapping between their elements. The elements could be images, patches, pixels, or texts under different contexts. With the emergence of deep neural networks, numerous problems, including learning correspondence, can be considered from the perspective of representation learning. In this thesis, we undertake the task of learning various types of correspondence and exploring their applications.

1.1 Spatial-Temporal Correspondence in Videos

A meaningful video correspondence characterizes the pixel motion between two consecutive frames. Typically, we use optical flow to represent the video motion, where a 2-dimensional vector depicts each pixel's horizontal and vertical movements. As shown in Figure 1.1, the optical flow visualizes the moving direction of every pixel in the input image. However, the pixel-level flow annotation is labor-intensive. There are a large number of unlabeled videos available, yet the pixel-level flow annotation is labor-intensive. Moreover, directly applying the pretrained supervised models on existing open datasets might not effectively address real-world problems. Because the dataset used for research only provides us with a limited number of labeled frames, which is far more insufficient for complex real-world scenes. Besides, a domain gap exists between the academic video datasets (including other synthetic labeled videos) and industrial



Input Image

Optical Flow

Color Key

Figure 1.1: Optical flow in real life scenes. Given the left input image, the optical flow is visualized in the middle image, and the right image is the color key map, where different color denotes different flow magnitudes and directions.

application scenarios. All of these motivate us to find an unsupervised flow estimation method that can use many unlabeled videos.

Numerous deep optical flow estimation methods are based on a similar neural network design: PWC-Net [112]. As shown in Figure 1.2, we generate L-level pyramids of feature representations, with the bottom (zeroth) level being the input images, and obtain the upper-level feature with a smaller resolution through a convolutional layer. At each level, we warp the second image's features toward the first image using the upsampled flow from the previous level. Next, we construct the cost volume by computing the correlation between features from the first image and the warped image. A cost volume stores the data matching cost for associating a pixel in the first image with surrounding pixels in the second image. Then, the optical flow estimator network is employed to estimate the flow at each level. Finally, a context network takes the upsampled flow and the second last layer from the optical flow estimator as inputs and outputs a refined flow.

Of all modules in the PWC-Net structure, the cost volume is the most prominent part of neural networks in other vision tasks. It plays a role in finding the correlation between the feature vector in the first frame and the potential feature vectors in the second frame. Here, we use the upsampled flow to find the possible corresponding feature vector in the second image. A rectangle around the corresponding feature vector bounds the potential feature vectors with horizontal and vertical replacements. The cost volume gives much

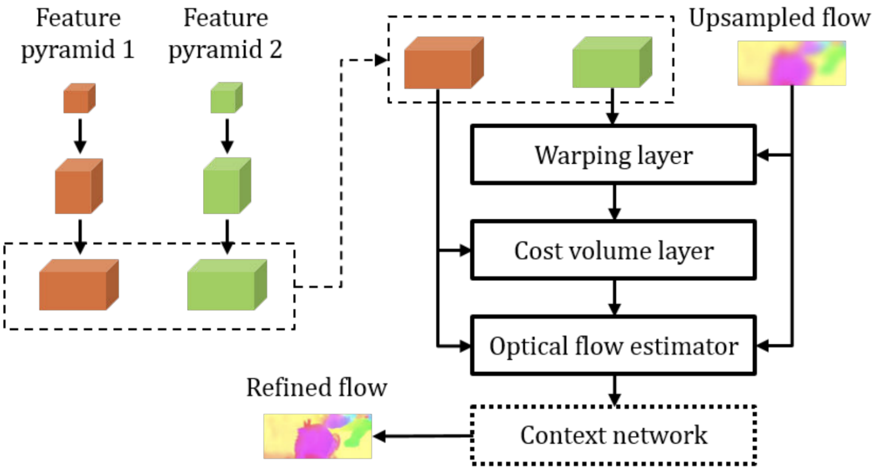


Figure 1.2: Network structure of PWC-Net.

correspondence information between two frames so that the following estimators could predict the optical flow well. However, the cost volume is constructed by computing the inner product of two feature vectors in the standard Euclidean inner product space, limiting the representation capacity of flow estimation models. Because the correlation among different channel dimensions is not considered, and each dimension contributes equally to the cost volume.

Chapter 2 proposes a method to learn better video correspondence by improving the cost volume. Specifically, we propose a learnable cost volume (LCV) layer using the elliptical inner product, which generalizes the standard inner product by introducing a learnable kernel weight matrix. To preserve the positive-definiteness of the kernel matrix during the training process, I perform the spectral decomposition on the kernel matrix and use the Cayley representation for re-parametrization. The learnable cost volume is a lightweight module and can be easily plugged into existing models to replace the vanilla cost volume. We show that the LCV module improves the accuracy of state-of-the-art models on standard benchmarks and the robustness against illumination change, noises, and adversarial perturbations of the input signals.

1.2 Semantic Correspondence in Images

In our second scenario, we delve into predicting pixel-level semantic correspondence between two different objects. For instance, as depicted in Figure 1.3, we recognize the tail of a fighter and civil aviation aircraft as a valid semantic correspondence despite originating from distinct objects. This task is notably more challenging than the video correspondence counterpart, primarily due to its requirement to establish pixel-level correspondences between two semantically similar objects. The increased complexity arises from several factors:

1. **Broader Scope:** Unlike the constrained context of video correspondence, image correspondence encompasses a more expansive domain. It extends beyond the confines of two consecutive frames within a single video and contains any two images housing potentially semantically similar objects.

2. **Global Correspondence:** In contrast to optical flow estimation, which restricts its search for matches within a defined displacement range, image correspondence mandates the identification of corresponding pairs on a global scale.

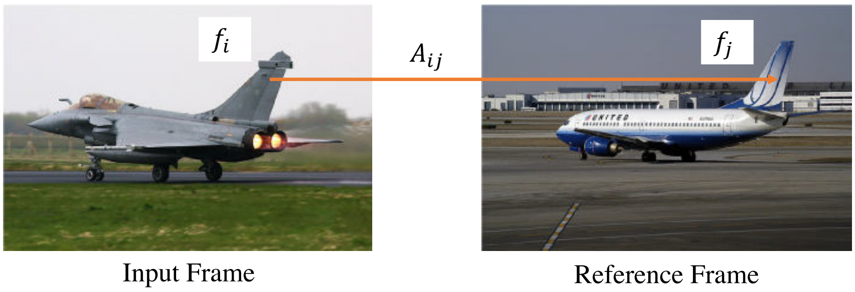


Figure 1.3: Illustration of an affinity matrix.

As shown in Figure 1.3, we construct the affinity matrix A by computing the similarity between pixel i from the input frame and pixel j from the reference frame as follows:

$$A_{ij} = \frac{\exp(f_i^\top f_j)}{\sum_k \exp(f_i^\top f_k)}, \tag{1.1}$$

where f_i and f_j are feature vectors from the input and reference frames. It could be easily seen that $\sum_j A_{ij} = 1$. Thus, each affinity matrix A column could be considered a probability distribution over all pixels in the reference frame. The affinity matrix contains the pixel-level correspondence information between input and reference frames, as the position of the maximum value in the j-th column can be considered as the correspondence for the pixel j in the input frame. Compared with a cost volume in the optical flow estimation, the advantage of an affinity matrix is that it contains correspondence information of all possible pairs rather than between a pixel and its neighbors.

The affinity matrix is also deeply interconnected with optical flow. Given two images I_1 and I_2 of the same shape $H \times W$, the affinity matrix of these two frames $A \in \mathbb{R}^{HW \times HW}$ can be obtained via Eq. (1.1). We can generate a grid matrix $M \in \mathbb{R}^{H \times W \times 2}$, where the (i, j)-th entry is a 2-dimensional vector indicating its position, i.e., (i, j). Then we reshape M into a flat grid matrix $G_1 \in \mathbb{R}^{HW \times 2}$. By multiplying the flat grid matrix with the affinity matrix, we can get the corresponding position of every pixel in I_2, denoted as G_2,

$$G_2 = AG_1. \tag{1.2}$$

Therefore, the optical flow from I_1 to I_2 is

$$F_{12} = G_2 - G_1 = (A - I)G_1 = -LG_1, \tag{1.3}$$

where $L := I - A$ is the Laplacian matrix. The formulae above suggest that the optical flow is closely related to the affinity matrix.

As the affinity is directly computed from two image features, the problem of learning image correspondence is the problem of representation learning. How do we learn good feature representations in the latent space so that semantically similar object parts from two images could be better matched in a self-supervised way?

In Chapter 3, we aim to address the problem of self-supervised representation learning for semantic image correspondence. As far as we know, most existing semantic correspondence methods design complicated matching algorithms based on deep features from pretrained ImageNet models. Their application in unlabeled data may be limited because pretrained ImageNet models are obtained by training on labeled images. We propose a model that combines momentum contrastive learning and image cycle learning to address this issue. Momentum contrastive learning aims to learn a discriminative global feature in a self-supervised way. In our proposed image cycle learning, we regard the correspondence relationship as a path connecting pixels between two images, and we traverse through all retrieved images cyclically. Our experimentation demonstrates a noticeable performance enhancement achieved by our method when evaluated on a standard benchmark dataset.

1.3 Multimodal Correspondence between Images and Texts

Vision-language models represent a significant advancement in the field of artificial intelligence, bridging the gap between visual and textual data. These models are designed to understand and generate meaningful connections between images and texts, enabling a wide range of applications, from image captioning and visual question-answering (VQA) to content recommendation and even aiding the visually impaired. They serve as the cornerstone for multimodal understanding, where multiple modes of data are integrated to extract richer, more comprehensive insights.

The core of vision-language models lies in their ability to establish meaningful correspondence between images and texts. The secret of establishing multi-modal correspondence lies in the training objective. Typically, a vision-language model consists of two encoders, where deep neural networks, especially transformers [121], are used to extract visual and textual representations. The training objective is to align image representations with text representations in the same embedding space by employing the contrastive loss, which is defined as follows,

$$L_{con} = -\frac{1}{N}\sum_{i}^{N}\log\frac{\exp(x_i^\top y_i/\sigma)}{\sum_{j=1}^{N}\exp(x_i^\top y_j/\sigma)} - \frac{1}{N}\sum_{i}^{N}\log\frac{\exp(y_i^\top x_i/\sigma)}{\sum_{j=1}^{N}\exp(y_i^\top x_j/\sigma)}, \tag{1.4}$$

where x_i and y_j are the normalized embedding of the image in the i-th pair and that of text in the j-th pair respectively, N is the batch size, and σ is the temperature to scale the logits. In addition to the image encoder, the dual-encoder approach also learns an aligned text encoder that enables cross-modal alignment applications such as image-text retrieval and zero-shot image classification.

Such multi-modal correspondence can be used in many applications.

1. **Image Captioning:** Given an image, the model generates a descriptive caption

that succinctly conveys the content and context of the image. For example, describing a photo of a beach scene with ”A sandy beach with palm trees and clear blue water.”

2. **Visual Question-Answering (VQA):** In VQA tasks, the model answers questions based on an image, demonstrating its ability to understand both visual content and textual input. For instance, when asked “What color is the boat in the picture?” the model can correctly respond with “blue.”

3. **Image Retrieval:** Vision-language models can be used to find images based on textual queries or vice versa. Given a sentence like “A red sports car,” the model can retrieve images that match this description.

4. **Text-to-Image Generation:** Conversely, these models can generate images from textual descriptions. If provided with the text “A cat sitting on a windowsill,” the model can create an image that corresponds to this description.

5. **Content Recommendation:** Vision-language models can also be employed to recommend relevant content based on user preferences and the understanding of both visual and textual content. For instance, suggesting videos or products based on a combination of textual reviews and images.

Chapter 4 delves into few-shot learning using large, pretrained vision-language models. Our focus centers on scenarios where only a limited number of images are available for downstream image classification tasks. In such cases, conventional performance often falls short in comparison to the zero-shot capabilities of these models. To address this, we harness the power of multi-modal correspondence to develop a more robust few-shot learning approach. Our approach involves computing text embeddings for the category names of the few-shot image classification datasets. These text embeddings are normalized and used to initialize the classification head. The effectiveness of this method is demonstrated across a wide range of datasets, offering a promising solution to the challenges posed by limited image data in the context of few-shot learning.

Chapter 2

Learnable Cost Volume Using the Cayley Representation

Cost volume is an essential component of recent deep models for optical flow estimation and is usually constructed by calculating the inner product between two feature vectors. However, the standard inner product in the commonly-used cost volume may limit the representation capacity of flow models because it neglects the correlation among different channel dimensions and weighs each dimension equally. To address this issue, we propose a *learnable cost volume* (LCV) using an elliptical inner product, which generalizes the standard inner product by a positive definite kernel matrix. To guarantee its positive definiteness, we perform spectral decomposition on the kernel matrix and re-parameterize it via the Cayley representation. The proposed LCV is a lightweight module and can be easily plugged into existing models to replace the vanilla cost volume. Experimental results show that the LCV module not only improves the accuracy of state-of-the-art models on standard benchmarks but also promotes their robustness against illumination change, noises, and adversarial perturbations of the input signals.

2.1 Introduction

Optical flow estimation is a fundamental computer vision task and has broad applications, such as video interpolation [3], video prediction [73], video segmentation [117, 13], and action recognition [74]. Despite the recent progress made by deep learning models, it is still challenging to accurately estimate optical flow for image sequences with large displacements, textureless regions, motion blur, occlusion, illumination changes, and non-Lambertian reflection.

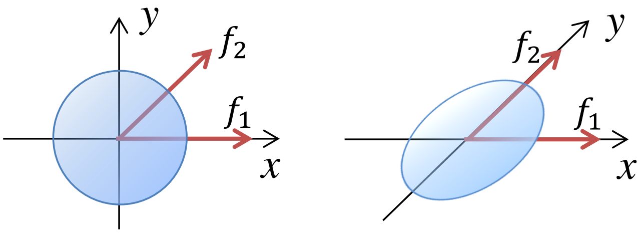


Figure 2.1: Standard inner product space v.s. elliptical inner product space.

Most deep optical flow models [112, 76, 46] adopt the idea of coarse-to-fine processing via feature pyramids and construct *cost volumes* at different levels of the pyramids. The cost volume stores the costs of matching pixels in the source image with their potential matching candidates in the target image. It is typically constructed by calculating the inner product between the convolutional features of one frame and those of the next frame, and then regressed to the estimated optical flow by an estimation sub-network. The accuracy of the estimated optical flow heavily relies on the quality of the constructed cost volume.

While the standard Euclidean inner product is widely used to build the cost volume (a.k.a., vanilla cost volume) for optical flow, we argue that it limits the representation capacity of the flow model for two reasons. First, the correlation among different channel dimensions is not taken into consideration by the standard Euclidean inner product. As shown in Fig. 2.1, we use a simple 2D example for illustration. Given two feature vectors $\boldsymbol{f}_1$ and $\boldsymbol{f}_2$ with positive correlation in the standard inner product space, we are able to find a proper elliptical inner product space to make these two feature vectors orthogonal to each other, which gives a zero correlation. Therefore, the specific choice of the inner product space influences the values of the matching costs, and thus should be further exploited. Second, each feature dimension contributes equally to the vanilla cost volume, which may give a sub-optimal solution to constructing the cost volume for flow estimation. Ideally, dimensions corresponding to noises and random perturbations should be suppressed, while those containing discriminative signals for flow estimation should be kept or magnified.

To address these limitations, we propose a *learnable cost volume* module which accounts for the correlation among different channel dimensions and re-weighs the contribution of each feature channel to the cost volume. The LCV generalizes the Euclidean inner product space to an elliptical inner product space, which is parameterized by a symmetric and positive definite kernel matrix. The spectral decomposition of the kernel matrix gives an orthogonal matrix and a diagonal matrix. The orthogonal matrix linearly transforms the features into a new feature space, which accounts for the correlation among different channel dimensions. The diagonal matrix multiplies each transformed feature by a positive scalar, which weighs each feature dimension differently. From a geometric perspective, the orthogonal matrix rotates the axes and the diagonal matrix stretches the axes so that the feature vectors are represented in a learned elliptical inner product space, which generates more discriminative matching costs for flow estimation.

However, directly learning a kernel matrix in an end-to-end manner cannot guarantee the symmetry and positive definiteness of the kernel matrix, which is required by the definition of inner product. To address this issue, we perform spectral decomposition on the kernel matrix and represent each component via the Cayley transform. Specifically, the special orthogonal matrices that exclude -1 as the eigenvalue can be bijectively mapped into the skew-symmetric matrices, and the diagonal matrices can be similarly represented by the composition of the Cayley transform and the arctangent function. In this way, all parameters of the learnable cost volume can be inferred in an end-to-end fashion without explicitly imposing any constraints.

The proposed learnable cost volume is a general version of the vanilla cost volume, and thus can replace the vanilla cost volume in the existing networks. We finetune the existing architectures equipped with LCV by initializing the kernel matrix as the identity matrix and restoring other parameters from the pre-trained models. Experimental results on the Sintel and KITTI benchmark datasets show that the proposed LCV significantly improves the performance of existing methods in both supervised and unsupervised settings. In addition, we demonstrate that LCV is able to promote the robustness of the existing models against illumination changes, noises, and adversarial attacks.

To summarize, we make the following contributions:

1. We propose a learnable cost volume to account for correlations among different feature dimensions and weight each dimension separately.

2. We employ the Cayley representation to re-parameterize the kernel matrix in a way that all parameters can be learned in an end-to-end manner.

3. The proposed LCV can easily replace the vanilla cost volume and improve the accuracy and robustness of the state-of-the-art models.

2.2 Related Work

2.2.1 Supervised Learning of Optical Flow

Inspired by the success of convolutional neural networks (CNNs) on per-pixel predictions such as semantic segmentation and single-image depth estimation, Dosovitski *et al.* propose FlowNet [22], the first end-to-end deep neural network capable of learning optical flow. FlowNet predicts a dense optical flow map from two consecutive image frames with an encoder-decoder architecture. FlowNet2.0 [50] extends FlowNet by stacking multiple basic FlowNet modules for iterative refinement and its accuracy is fully on par with those of the state-of-the-art methods at the time. Motivated by the idea of coarse-to-fine refinement in traditional optical flow methods, SpyNet [99] introduces a compact spatial pyramid network that warps images at multiple scales to deal with displacements caused by large motions. PWC-Net [112] extracts features through pyramidal processing and builds a cost volume at each level from the warped and the target features to iteratively refine the estimated flow. VCN [137] improves the cost volume processing by decoupling the 4D convolution into a 2D spatial filter and a 2D winner-take-all (WTA) filter, while still retaining a large receptive field. HD^3 [139] learns a probabilistic matching density distribution at each scale and merges the matching densities at different scales to recover the global matching density.

2.2.2 Unsupervised Learning of Optical Flow

The advantage of unsupervised methods is that it can sidestep the limitations of the synthetic datasets and exploit the large number of training data in the realistic domain. In [53] and [102], the flow guidance comes from warping the target image according to the predicted flow and comparing against the reference image. The photometric loss is adopted to ensure brightness constancy and spatial smoothness. In some work [130, 82], occluded regions are excluded from the photometric loss. As pixels occluded in the target image are also absent in the warped one, enforcing matching of the occluded pixels would misguide the training. Wang *et al.* [130] obtain an occlusion mask from the range map inferred from the backward flow, while UnFlow [82] relies on the forward-backward consistency to estimate the occlusion mask. Unlike these two methods that predict the occlusion map in advance with certain heuristic, Back2Future [52] estimates the occlusion and optical flow jointly by introducing a multi-frame formulation and reasoning the occlusion in a more advanced manner. DDFlow [76] performs knowledge distillation by cropping patches from the unlabeled images, which provides flow guidance for the occluded regions. SelFlow [77] hallucinates synthetic occlusions by perturbing super-pixels where the occluded regions are guided by a model pre-trained from non-occluded regions.

2.2.3 Correspondence Matching

Typically, stereo matching algorithms [106, 43] involve local correspondence extraction and smoothness regularization, where the smoothness regularization is enforced by energy minimization. Recently, hand-crafted features are replaced by deep features and minimization of the matching cost is substituted by training convolutional neural networks [143, 59]. Xu *et al.* [136] construct a 4D cost volume using an adaptation of the semi-global matching, and Yang *et al.* [137] reduce the computation overhead of processing the 4D matching volume by factorizing into two separable filters.

Different from these approaches where the correspondence is represented by a hand-crafted matching cost volume, we propose a learnable cost volume that can capture the correlation among different channels by adapting the features to an elliptical inner product space. Such a correlation is automatically learned by optimizing the kernel matrix using the Cayley representation, which is more flexible and effective in optical flow estimation and can be easily plugged into the existing architectures. To our knowledge, this chapter is the first one to use the Cayley representation for learning correspondence in optical flow.

2.3 Learnable Correlation Volume

2.3.1 Vanilla Cost Volume

Let $\boldsymbol{F}^1, \boldsymbol{F}^2 \in \mathbb{R}^{c \times h \times w}$ be the convolutional feature of the first frame and the warped feature of the second frame, respectively. The vanilla cost volume is defined as the inner product between the query feature $\boldsymbol{F}^1_{i,j}$ and the potential match candidate $\boldsymbol{F}^2_{k',l'}$, *i.e.,*

$$C(\boldsymbol{F}^1, \boldsymbol{F}^2)_{k,l,i,j} = \boldsymbol{F}^{1\top}_{i,j} \boldsymbol{F}^2_{k',l'}, \tag{2.1}$$

which maps from the space $\mathbb{R}^{c \times h \times w} \times \mathbb{R}^{c \times h \times w}$ to $\mathbb{R}^{u \times v \times h \times w}$. Here, u and v are usually odd numbers, indicating the displacement ranges in horizontal and vertical directions, (i, j) denotes the spatial location of the feature map $\boldsymbol{F}^1$, and $(k', l') = (i - (u-1)/2 + k, j - (v-1)/2 + l)$ denotes that of $\boldsymbol{F}^2$. For each location (i, j) of the query feature $\boldsymbol{F}^1$, the matching is performed against pixels of $\boldsymbol{F}^2$ within a $u \times v$ search window centered by the location (i, j). Then, the cost volume is either reshaped into $uv \times h \times w$ and post-processed by 2D convolutions [112], or kept as a 4D tensor on which the separable 4D convolutions [137] are applied.

2.3.2 Learnable Cost Volume

We generalize the standard Euclidean inner product to the elliptical inner product, where the matching cost is computed as follows:

$$C(\boldsymbol{F}^1, \boldsymbol{F}^2)_{k,l,i,j} = \boldsymbol{F}^{1\top}_{i,j} \boldsymbol{W} \boldsymbol{F}^2_{k',l'}. \tag{2.2}$$

Here, $\boldsymbol{W} \in \mathbb{R}^{c \times c}$ is a learnable kernel matrix that determines the elliptical inner product space, and other notations are the same as those in Eq. (2.1). According to the definition of inner product, $\boldsymbol{W}$ should be a symmetric and positive definite matrix. By spectral decomposition, we obtain

$$\boldsymbol{W} = \boldsymbol{P}^\top \Lambda \boldsymbol{P}, \tag{2.3}$$

where $\boldsymbol{P}$ is an orthogonal matrix, and Λ is a diagonal matrix with positive entries, *i.e.,* $\Lambda = \mathrm{diag}(\lambda_1, \cdots, \lambda_c)$ with $\lambda_i > 0,\ \forall i \in \{1, \cdots, c\}$. The orthogonal matrix $\boldsymbol{P}$ actually rotates the coordinate axes and the diagonal matrix Λ re-weights different dimensions, which directly address the two limitations mentioned in Sec. 2.1.

2.3.3 Learning with the Cayley Representation

In the proposed LCV module, the entries of the kernel matrix $\boldsymbol{W}$ are the only learnable parameters. However, the constraints of symmetry and positive-definiteness hinders the gradient-based end-to-end learning of $\boldsymbol{W}$. To address this issue, we propose to optimize $\boldsymbol{P}$ and Λ instead of $\boldsymbol{W}$.

One way to optimize $\boldsymbol{P}$ is to employ the Riemann gradient descent on the Stiefel manifold, which is defined as

$$V_k(\mathbb{R}^n) = \{\boldsymbol{A} \in \mathbb{R}^{n\times k} | \boldsymbol{A}^\top \boldsymbol{A} = \boldsymbol{I}_k\}. \tag{2.4}$$

All orthogonal matrices lie in the Stiefel manifold. Specifically, $\boldsymbol{P} \in V_c(\mathbb{R}^c)$. Therefore, we can apply the Riemann gradient descent on the Stiefel matrix manifold, where the projection and retraction formula [1] are given by

$$\mathcal{P}_{\boldsymbol{X}}(\boldsymbol{Z}) = (\boldsymbol{I} - \boldsymbol{X}\boldsymbol{X}^\top)\boldsymbol{Z} + \boldsymbol{X} \cdot \mathrm{skew}(\boldsymbol{X}^\top \boldsymbol{Z}) \tag{2.5}$$

$$\mathcal{R}_{\boldsymbol{X}}(\boldsymbol{Z}) = (\boldsymbol{X} + \boldsymbol{Z})(\boldsymbol{I} + \boldsymbol{Z}^\top \boldsymbol{Z})^{-\frac{1}{2}}, \tag{2.6}$$

where $\mathrm{skew}(\boldsymbol{X}) := (\boldsymbol{X} - \boldsymbol{X}^\top)/2$. However, to perform the Riemann gradient descent, the projection and retraction operations are required in each training step, and the matrix multiplication brings considerable computational overhead.

We can address this issue in a more elegant way using the Cayley Representation [8]. First, we define a set of matrices:

$$\mathrm{SO}^*(n) := \{\boldsymbol{A} \in \mathrm{SO}(n) : -1 \notin \sigma(\boldsymbol{A})\}, \tag{2.7}$$

where $\sigma(\boldsymbol{A})$ denotes the spectrum, *i.e.,* all eigenvalues, of $\boldsymbol{A}$. $\mathrm{SO}^*(n)$ is a subset of the special orthogonal group $\mathrm{SO}(n)$ and the spectrum of its elements excludes -1. Then, we have the following theorems:

Theorem 1 (Cayley Representation)**.** *Given any matrix* $\boldsymbol{P} \in \mathrm{SO}^*(n)$*, there exists a unique skew-symmetric matrix* $\boldsymbol{S}$*,* i.e., $\boldsymbol{S}^\top = -\boldsymbol{S}$*, such that*

$$\boldsymbol{P} = (\boldsymbol{I} - \boldsymbol{S})(\boldsymbol{I} + \boldsymbol{S})^{-1}. \tag{2.8}$$

Theorem 2. *The set of matrices* $\mathrm{SO}^*(n)$ *is connected.*

By Theorem 1, we can initialize the matrix $\boldsymbol{P}$ in Eq. (2.3) as an identity matrix $\boldsymbol{I} \in \mathrm{SO}^*(c)$, and update $\boldsymbol{S}$ so as to update $\boldsymbol{P}$ using gradient-based optimizer. Let $\boldsymbol{P}^*$ be the optimal orthogonal matrix, and we claim that it is possible to reach $\boldsymbol{P}^*$ from initializing as the identity matrix $\boldsymbol{P} = \boldsymbol{I}$. This because $\mathrm{SO}^*(c)$ is a connected set (Theorem 2), so there exists a continuous path joining $\boldsymbol{I} \in \mathrm{SO}^*(c)$ and any $\boldsymbol{P} \in \mathrm{SO}^*(c)$, including $\boldsymbol{P}^*$.

Due to the positive definiteness of $\boldsymbol{W}$, the constraint of the diagonal matrix $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_c)$ is $\lambda_i > 0,\ \forall i = 1, \ldots, c$. Thus, we map $\mathbb{R}$ to $\mathbb{R}^+$ by applying the composition of the Cayley transform and the arctangent function, *i.e.,*

$$\lambda_i = \frac{\pi + 2 \arctan t_i}{\pi - 2 \arctan t_i}, \tag{2.9}$$

where $t_i \in \mathbb{R}$ is free of constraint.

The above re-parameterization trick enables us to update the kernel matrix $\boldsymbol{W}$ in an end-to-end manner using the SGD optimizer or its variants, which alleviates the heavy computation brought by the projection and retraction and makes the training process much easier.

2.3.4 Interpretation

To better understand the learnable cost volume, we analyze several cases here.

1. $\boldsymbol{W} = \boldsymbol{I}$**.**

This degenerates into the vanilla cost volume, in which the standard Euclidean inner product is adopted.

2. $\boldsymbol{W} = \Sigma^{-1}$.

Let Σ be the covariance matrix of the convolutional feature, then the learnable cost volume is essentially a whitening transformation. Let $\boldsymbol{Q} = \Lambda^{1/2}\boldsymbol{P}$, and then Eq. (2.2) can be formulated as

$$C(\boldsymbol{F}^1, \boldsymbol{F}^2)_{k,l,i,j} = \boldsymbol{F}^{1\top}_{i,j}\boldsymbol{P}^\top\Lambda^{1/2}\Lambda^{1/2}\boldsymbol{P}\boldsymbol{F}^2_{k',l'} = (\boldsymbol{Q}\boldsymbol{F}^1_{i,j})^\top(\boldsymbol{Q}\boldsymbol{F}^2_{k',l'}), \tag{2.10}$$

where $\boldsymbol{Q}\boldsymbol{F}^1_{i,j}$ represents the transformed feature of $\boldsymbol{F}^1_{i,j}$ after PCA [56] whitening. Similarly, letting $\boldsymbol{R} = \boldsymbol{P}^\top\Lambda^{1/2}\boldsymbol{P}$, we can have

$$C(\boldsymbol{F}^1, \boldsymbol{F}^2)_{k,l,i,j} = \boldsymbol{F}^{1\top}_{i,j}\boldsymbol{P}^\top\Lambda^{1/2}\boldsymbol{P}\boldsymbol{P}^\top\Lambda^{1/2}\boldsymbol{P}\boldsymbol{F}^2_{k',l'} = (\boldsymbol{R}\boldsymbol{F}^1_{i,j})^\top(\boldsymbol{R}\boldsymbol{F}^2_{k',l'}), \tag{2.11}$$

where $\boldsymbol{R}\boldsymbol{F}^1_{i,j}$ is the transformed feature of $\boldsymbol{F}^1_{i,j}$ after ZCA [6] whitening. It has been shown that the high-level styles can be removed with the contextual structures remained by whitening the convolutional features [72].

3. $\boldsymbol{W} = \boldsymbol{P}^\top\Lambda\boldsymbol{P}$.

The learnable cost volume shares a similar formula as the whitening process, but $\boldsymbol{W}$ is learned over the whole training dataset rather than statistics of two inputs, thus contains certain holistic information of the entire training dataset. Because it has been verified that the certain holistic characteristics of the underlying image can be captured by the Gram matrix along the channel dimension [27, 72]. The learnable cost volume performs as "whitening" features using the common information learned from all frames. Specifically, the orthogonal matrix $\boldsymbol{P}$ re-arranges the information across the channel dimension, while the diagonal matrix Λ filters out insignificant signals, making the correlation more robust to the illumination changes and noises. (See Sec. 2.4.4.)

It should also be pointed out that the whitening matrix $\boldsymbol{R}$ in Eq. (2.11) could be viewed as a 1 × 1 conv functioning on the feature, but directly applying a 1 × 1 conv with learnable parameters on features before computing the standard cost volume cannot replace the proposed learned cost volume. Because $\boldsymbol{R}^\top\boldsymbol{R}$ only gives a positive semi-definite matrix even when $\boldsymbol{R}$ is full-rank, which does not meet the positive definiteness property of an inner product.

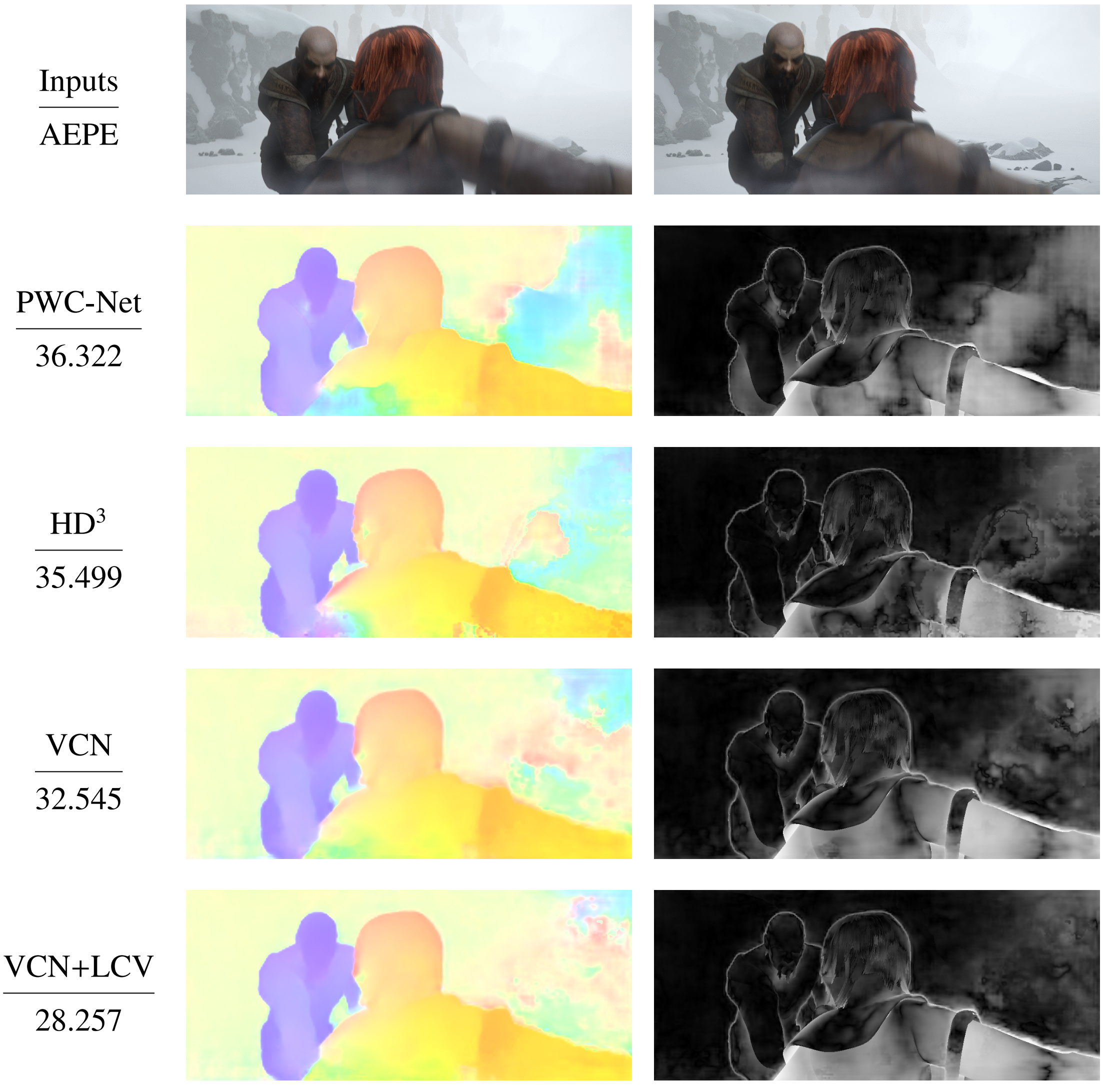


Figure 2.2: Visual results on “Ambush 1” from the Sintel test final pass. The number under each method denotes the average end-point error (AEPE). Left: estimated flow; right: error map (increases from black to white).

2.3.5 Relation with the Weighted Sum of Squared Difference

The learnable cost volume can be also formulated by re-thinking the simplest matching criterion for comparing two features, *i.e.,* the weighted sum of squared difference

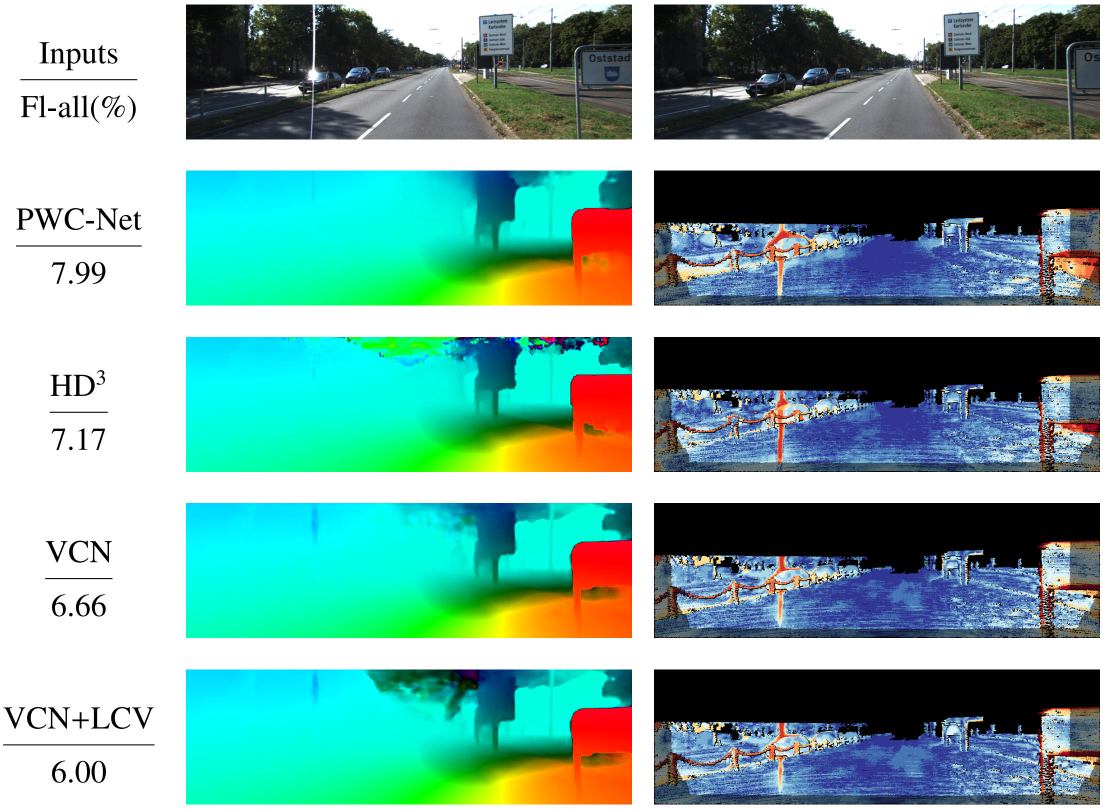


Figure 2.3: Visual results on the KITTI 2015 test set. The number under each method name denotes the Fl-all score on the given frames. Left: estimated flow; right: error map (increases from blue to red).

(WSSD):

$$\sum_i \lambda_i \left(G_i(\boldsymbol{F}^2) - G_i(\boldsymbol{F}^1) \right)^2, \tag{2.12}$$

where $G : \mathbb{R}^c \to \mathbb{R}^c$ denotes a transformation function on the features $\boldsymbol{F}^i \in \mathbb{R}^c,\ i = 1, 2$, and $G_i(\boldsymbol{F})$ indicates the i^{th} element of $G(\boldsymbol{F})$.

By the Taylor series expansion, we have

$$\sum_i \lambda_i \left(G_i(\boldsymbol{F}^2) - G_i(\boldsymbol{F}^1) \right)^2 \approx \sum_i \lambda_i \left(\nabla G_i(\boldsymbol{F}^1)^\top \Delta \boldsymbol{F} \right)^2 = \Delta \boldsymbol{F}^\top \boldsymbol{W} \Delta \boldsymbol{F}, \tag{2.13}$$

where $\Delta\boldsymbol{F}$=$\boldsymbol{F}^2 - \boldsymbol{F}^1$ is the feature difference and $\boldsymbol{W} = \sum_i \lambda_i \nabla G_i(\boldsymbol{F}^1) \nabla G_i(\boldsymbol{F}^1)^\top$ is the auto-correlation matrix. Here, $\boldsymbol{W}$ coincides with the kernel matrix of the proposed LCV module in Eq. (2.2). When $\lambda_i = 1 (i = 1, \ldots, c)$ and G is an identity map, then $\boldsymbol{W} = \boldsymbol{I}$,

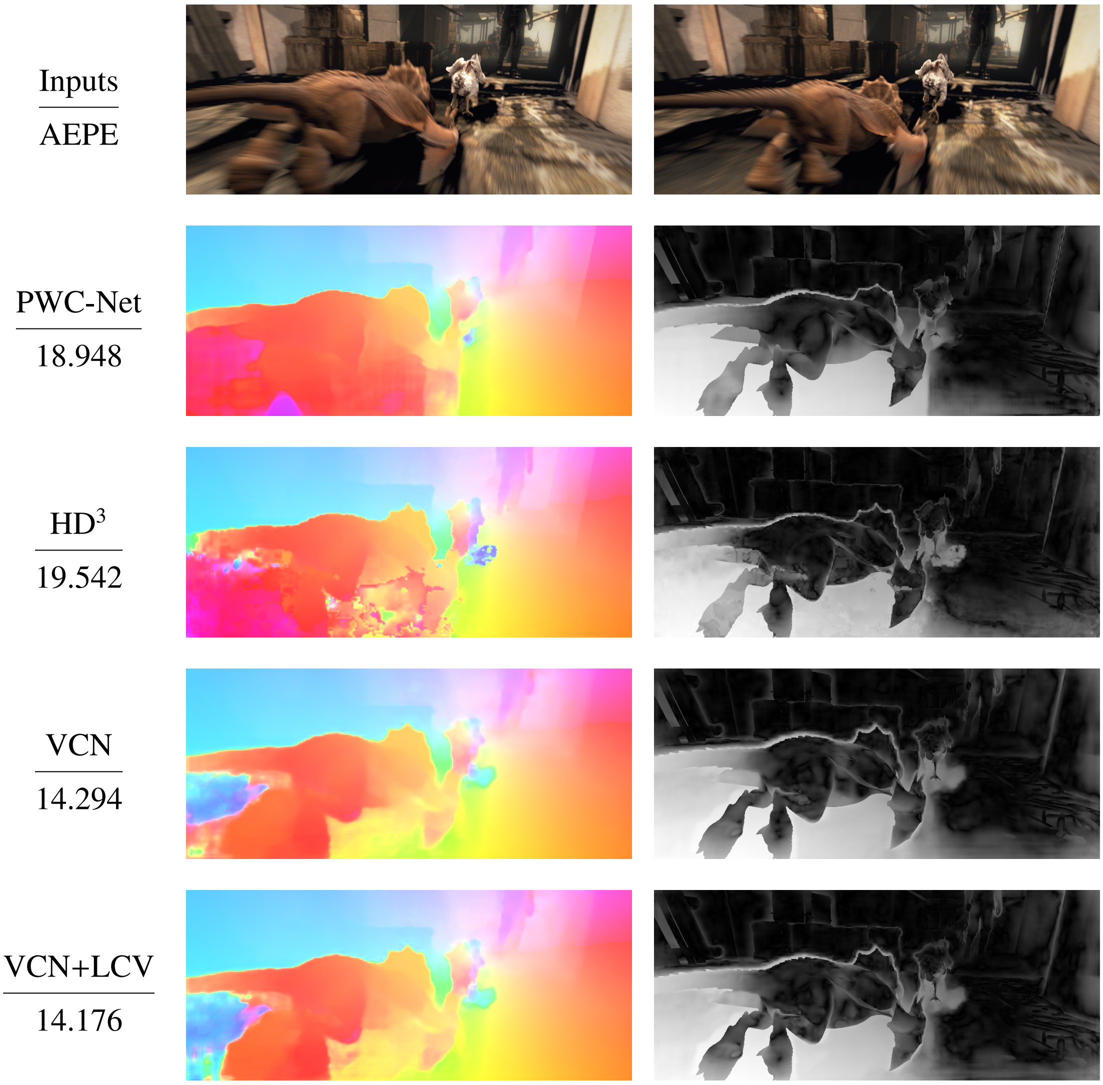


Figure 2.4: More visualization results on “Market 4” from the Sintel test final pass. The number under each method name denotes the average end-point error (AEPE) on the given frames. The estimated flow and error maps are presented on the left and right sides, respectively. In the error map, the error of the estimated flow increases from black to white.

which corresponds to the vanilla cost volume. If we further expand Eq. (2.13), we can

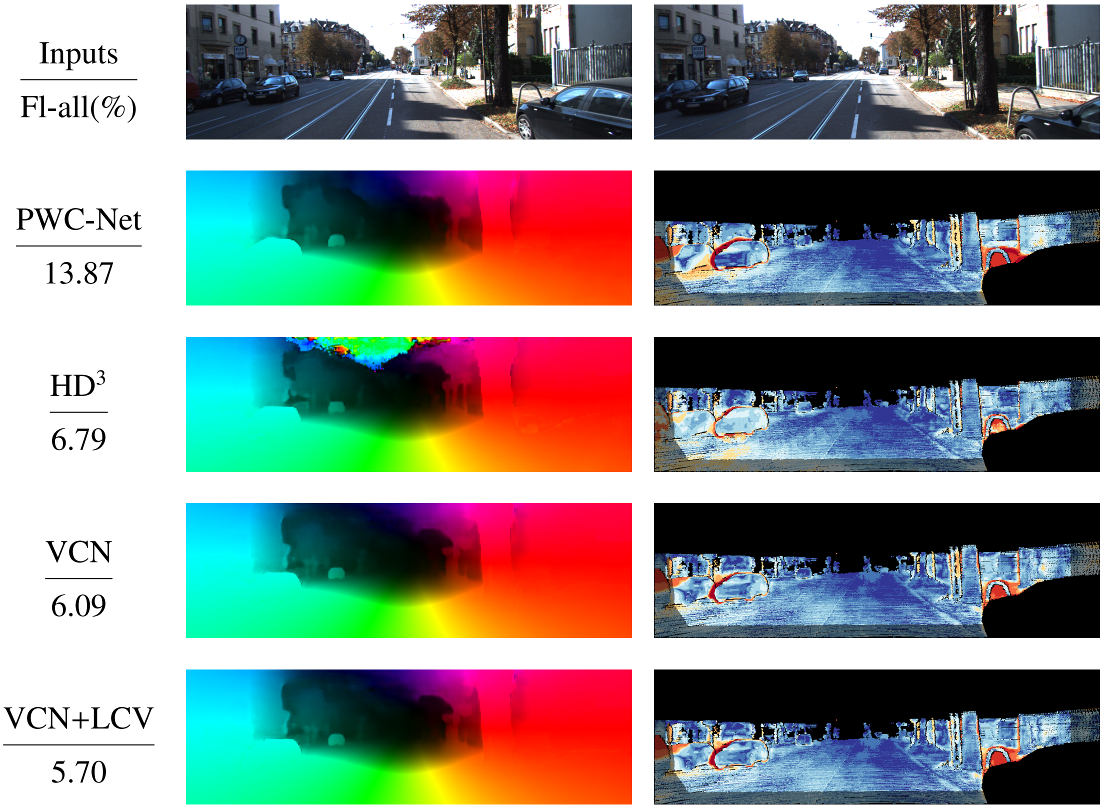


Figure 2.5: More visualization results on the KITTI 2015 test set. The number under each method name denotes the Fl-all score on the given frames. The estimated flow and error maps are presented on the left and right sides, respectively. From blue to red, the error of the estimated flow increases in the error map.

see the connection with the proposed learnable correlation volume as follows:

$$\begin{aligned} \Delta \boldsymbol{F}^\top \boldsymbol{W} \Delta \boldsymbol{F} &= (\boldsymbol{F}^2 - \boldsymbol{F}^1)^\top \boldsymbol{W} (\boldsymbol{F}^2 - \boldsymbol{F}^1) \\ &= (\boldsymbol{F}^{2\top} \boldsymbol{W} \boldsymbol{F}^2 + \boldsymbol{F}^{1\top} \boldsymbol{W} \boldsymbol{F}^1) - 2\boldsymbol{F}^{1\top} \boldsymbol{W} \boldsymbol{F}^2, \end{aligned} \tag{2.14}$$

where the last term shares the same formula with the proposed learnable cost volume. This implies that the proposed learnable cost volume is inversely correlated with WSSD. As WSSD measures the discrepancy between two features, the learnable cost volume characterizes a certain kind of similarity between them.

Table 2.1: Results of the supervised methods on the MPI Sintel and KITTI 2015 optical flow benchmarks. All reported numbers indicate the average endpoint error (AEPE) except for the last two columns, where the percentage of outliers averaged over all groundtruth pixels (Fl-all) are presented. "-ft" means finetuning on the relative MPI Sintel or KITTI training set and the numbers in the parenthesis are results that train and test on the same dataset. Missing entries (-) indicate that the results are not reported for the respective method. The best result for each metric is printed in bold.

Methods	Sintel				KITTI 2015		
	Clean		Final		AEPE	Fl-all (%)	
	train	test	train	test	train	train	test
FlowNet2 [50]	2.02	3.96	3.14	6.02	10.06	30.37	-
FlowNet2-ft [50]	(1.45)	4.16	(2.01)	5.74	(2.30)	(8.61)	10.41
DCFlow [136]	-	3.54	-	5.12	-	15.09	14.83
MirrorFlow [48]	-	-	-	6.07	-	9.93	10.29
SpyNet [99]	4.12	6.69	5.57	8.43	-	-	-
SpyNet-ft [99]	(3.17)	6.64	(4.32)	8.36	-	-	35.07
LiteFlowNet [46]	2.52	-	4.05	10.39	-	-	-
LiteFlowNet+ft [46]	(1.64)	4.86	(2.23)	6.09	(2.16)	-	10.24
PWC-Net [112]	2.55	-	3.93	-	10.35	33.67	-
PWC-Net-ft [112]	(2.02)	4.39	(2.08)	5.04	(2.16)	(9.80)	9.60
PWC-Net+-ft [113]	(1.71)	3.45	(2.34)	4.60	(1.50)	(5.30)	7.72
IRR-PWC-ft [49]	(1.92)	3.84	(2.51)	4.58	(1.63)	(5.30)	7.65
HD^3 [139]	3.84	-	8.77	-	13.17	23.99	-
HD^3-ft [139]	(1.70)	4.79	(1.17)	4.67	(1.31)	(4.10)	6.55
VCN [137]	2.21	-	3.62	-	8.36	25.10	8.73
VCN-ft [137]	(1.66)	2.81	(2.24)	4.40	(1.16)	(4.10)	6.30
RAFT [115]	1.09	2.77	1.53	3.61	(1.07)	(3.92)	6.30
RAFT (warm start) [115]	1.10	**2.42**	1.61	3.39	-	-	-
VCN+LCV	(1.62)	2.83	(2.22)	4.20	(1.13)	(3.80)	**6.25**
RAFT+LCV	**(0.94)**	2.75	**(1.31)**	3.55	**(1.06)**	**(3.77)**	6.26
RAFT+LCV (warm start)	(0.99)	2.49	(1.47)	**3.37**	-	-	-

2.4 Experiments

In this section, we present the experimental results of optical flow estimation in both supervised and unsupervised settings to demonstrate the effectiveness of the proposed learnable cost volume. Also, we carry out ablation studies to show that the LCV module performs favorably against other counterparts. Moreover, we analyze the behavior of LCV and find it beneficial to handling three challenging cases. More results can be found in the supplementary material and the source code and trained models is available at `https://github.com/Prinsphield/LCV`.

Training Process. It is well-known that the deep optical flow estimation pipeline consists the following stages in the supervised settings [113]: 1) train the model on the FlyingChairs [22] dataset; 2) finetune the model on the FlyingThings3D [81] dataset; and 3) finetune the model on the Sintel [7] and KITTI [84, 83] training sets. Besides, there are lots of tricks such as data augmentation and learning rate disruption, making the training process more complicated.

To avoid the tedious training procedure over multiple datasets, we adopt a more efficient way to train the model equipped with LCV. As mentioned in Sec. 2.3.4, the vanilla cost volume is a special case of the learnable cost volume when $\boldsymbol{W} = \boldsymbol{I}$, which means that the learnable cost volume is more general and backward compatible with vanilla cost volume. Therefore, we initialize the kernel matrix $\boldsymbol{W}$ as the identity matrix and other parameters are directly restored from the pre-trained models without using LCV. After that, we finetune the model with LCV on the Sintel or KITTI datasets using the same loss function. This training process not only significantly reduces training time but also plays a crucial role in the success under the unsupervised settings. (See Sec. 2.4.2.) This approach can also be viewed as fixing the kernal matrix as $\boldsymbol{W} = \boldsymbol{I}$ in the first three training stages, and let $\boldsymbol{W}$ be learnable in the final stage.

2.4.1 Supervised Optical Flow Estimation

First, we incorporate the learnable cost volume in the VCN [137] and RAFT [115] framework, and compare them with other existing methods. As shown in Table 2.1, our method performs favorably against other state-of-the-art methods on the Sintel Clean/Final pass and the KITTI 2015 benchmark.

The proposed LCV module improves the performance of VCN and RAFT by transforming the features of video frames to a whitened space to obtain a clean and robust matching correlation. This could account for the performance improvement on the Sintel Final pass, where the scenarios are much harder. As shown in Fig. 2.2, the flow estimation error for the snow background at the right side is smaller than other methods. This is a challenging case because the front person's arm renders occlusion to part of the snow background and the background is nearly all white, providing few clues for matching. However, the LCV module exploits more information from the correlation among different channels, which assists in obtaining the coherent flow estimation in the snow background. The LCV module also has an edge over the vanilla cost volume under the circumstance of light reflection and occlusion. As shown in Fig. 2.3, the prediction error of our method is smaller around the light reflection region and the rightmost traffic sign. The flow boundary of the dragon in Fig. 2.4 predicted by VCN+LCV is better than the other methods and the flow prediction near the tree (in front of the car) and fence by our method in Fig. 2.5 is more accurate compared with those of the others. The flow prediction for these pixels are are challenging due to the occlusion. LCV explores more information among channel dimensions, which could help alleviate the problem of occlusion to some extent.

Although we do not report the model parameters in the table, the proposed LCV module only makes a very slight increase in the model size. The additional parameters come from the kernel matrices $\boldsymbol{W} \in \mathbb{R}^{c \times c}$ at different pyramid levels. Taking VCN+LCV as an example, there are five kernel matrices in total, whose channel dimensions are 64, 64, 128, 128, and 128, respectively. The LCV module only takes up $64^2 \times 2 + 128^2 \times 3 = 57,344$ parameters, which is negligible compared with the entire VCN model of around 6.23M parameters.

2.4.2 Unsupervised Optical Flow Estimation

We also test the LCV module in unsupervised settings on the KITTI 2015 benchmark. We replace the vanilla cost volume with the LCV module in the DDFlow [76] model, and compare it with other unsupervised methods. As shown in Table 2.2, our model outperforms the DDFlow baseline, and even performs favorably against SelFlow [77], an improved version of DDFlow.

The training process is crucial to the success of the LCV module in the unsupervised methods. Different from the supervised training of optical flow models, there is no ground truth for direct supervision. Instead, most unsupervised methods use the photometric loss as a proxy loss. Specifically, the training of DDFlow consists of two stages: 1) pre-train a non-occlusion model with census transform [31], and 2) train an occlusion model by distillation from the non-occlusion model. If we directly follow the same procedure, the training of DDFlow+LCV will run into trivial solutions, as the photometric loss does not give a strong supervision for the correspondence learning, especially when the LCV module increases the dimension of the solution space. To prevent from trivial solutions, we fix the kernel matrix as $\boldsymbol{W} = \boldsymbol{I}$ in the pre-train stages, and update $\boldsymbol{W}$ in the distillation stage.

Table 2.2: Results of the unsupervised methods on the KITTI 2015 optical flow benchmark. Missing entries (-) indicate that the results are not reported for the respective method. The best result for each metric is printed in bold.

Methods	KITTI 2015			
	train	test		
	AEPE	Fl-bg (%)	Fl-fg (%)	Fl-all (%)
DSTFlow [102]	16.79	-	-	39
GeoNet [140]	10.81	-	-	-
UnFlow [82]	8.88	-	-	28.95
DF-Net [156]	7.45	-	-	22.82
OccAwareFlow [130]	8.88	-	-	31.20
Back2FutureFlow [52]	6.59	22.67	24.27	22.94
SelFlow [77]	**4.84**	**12.68**	21.74	14.19
DDFlow [76]	5.72	13.08	20.40	14.29
DDFlow+LCV (Ours)	5.15	12.98	**19.83**	**14.12**

2.4.3 Ablation Study

We evaluate multiple variants of the LCV module based on the VCN baseline:

- VCN: the original VCN baseline.
- VCN (ct): continue training the existing VCN using a small learning rate for more epochs.
- VCN ($\boldsymbol{W}$, ct): remove the symmetry and positive definiteness constraint of $\boldsymbol{W}$, *i.e.,* , not using the Cayley representation. We restore the weights from the pre-trained VCN and continue training the model with free $\boldsymbol{W}$.
- VCN ($\boldsymbol{\Lambda}$, ct): fix $\boldsymbol{P}$ to be an identity matrix and make the diagonal matrix $\boldsymbol{\Lambda}$ learnable.
- VCN ($\boldsymbol{P}$, ct): fix $\boldsymbol{\Lambda}$ to be an identity matrix and make the orthogonal matrix $\boldsymbol{P}$ learnable.
- VCN (1x1 conv): replace the positive definite $\boldsymbol{W}$ with $\boldsymbol{R}^\top\boldsymbol{R}$, where $\boldsymbol{R}$ is a 1×1 conv operating on features with input and output dimensions equal. $\boldsymbol{R}^\top\boldsymbol{R}$ is only a positive semi-definite matrix.
- VCN+LCV: employ the Cayley representation to ensure the symmetry and positive definiteness of $\boldsymbol{W}$.

We randomly split the 200 images with ground truth from the KITTI 2015 training set into the training and validation set by a ratio of 4:1. As shown in Table 2.3, we report the AEPE/Fl-all scores on the validation set. We observe that continuing training of the VCN model does not bring any benefit, which indicates that the best VCN model is not obtained at the very end of the training. Another interesting observation is that VCN ($\boldsymbol{W}$, ct) performs better than VCN (ct), showing the benefit of increasing the model capacity. However, it does not outperform VCN, not even VCN+LCV, confirming the importance of using a valid inner product space. Comparing the result of VCN (1x1 conv), we can further conclude that ensuring the positive definiteness via the Cayley representation is crucial to the performance. We can also find that VCN ($\boldsymbol{\Lambda}$, ct) gets a lower AEPE and VCN ($\boldsymbol{P}$, ct) gets a lower Fl-all compared with vanilla VCN. VCN+LCV combines the advantages of both axis rotation and re-weighting, aiming to address two limitations mentioned in the chapter.

Table 2.3: Ablation study of different variants of VCN on the KITTI 2015 dataset.

Methods	VCN	VCN (ct)	VCN ($\boldsymbol{W}$, ct)	VCN ($\boldsymbol{\Lambda}$, ct)	VCN ($\boldsymbol{P}$, ct)	VCN(1x1 conv)	VCN + LCV
AEPE/Fl-all	3.9/1.144	4.2/1.204	4.1/1.193	3.8/1.136	3.9/1.129	3.9/1.163	3.8/1.132

Table 2.4: Results on three challenging cases (numbers: AEPE/Fl-all scores).

(a) Illumination change

γ	0.2	0.3	0.4	0.5	0.7	1.0	2.0	3.0
VCN	16.8/3.240	9.9/1.891	5.9/1.306	3.8/0.995	2.7/0.834	2.5/0.805	2.6/0.819	2.6/0.826
VCN+LCV	17.1/3.232	9.8/1.866	5.9/1.273	3.7/0.967	2.6/0.804	2.4/0.775	2.4/0.790	2.5/0.804

(b) Noise

Standard deviation	0.0001	0.001	0.01	0.1
VCN	2.6/0.816	2.9/0.868	5.0/1.157	19.6/3.213
VCN+LCV	2.4/0.785	2.7/0.838	4.7/1.107	18.9/3.043

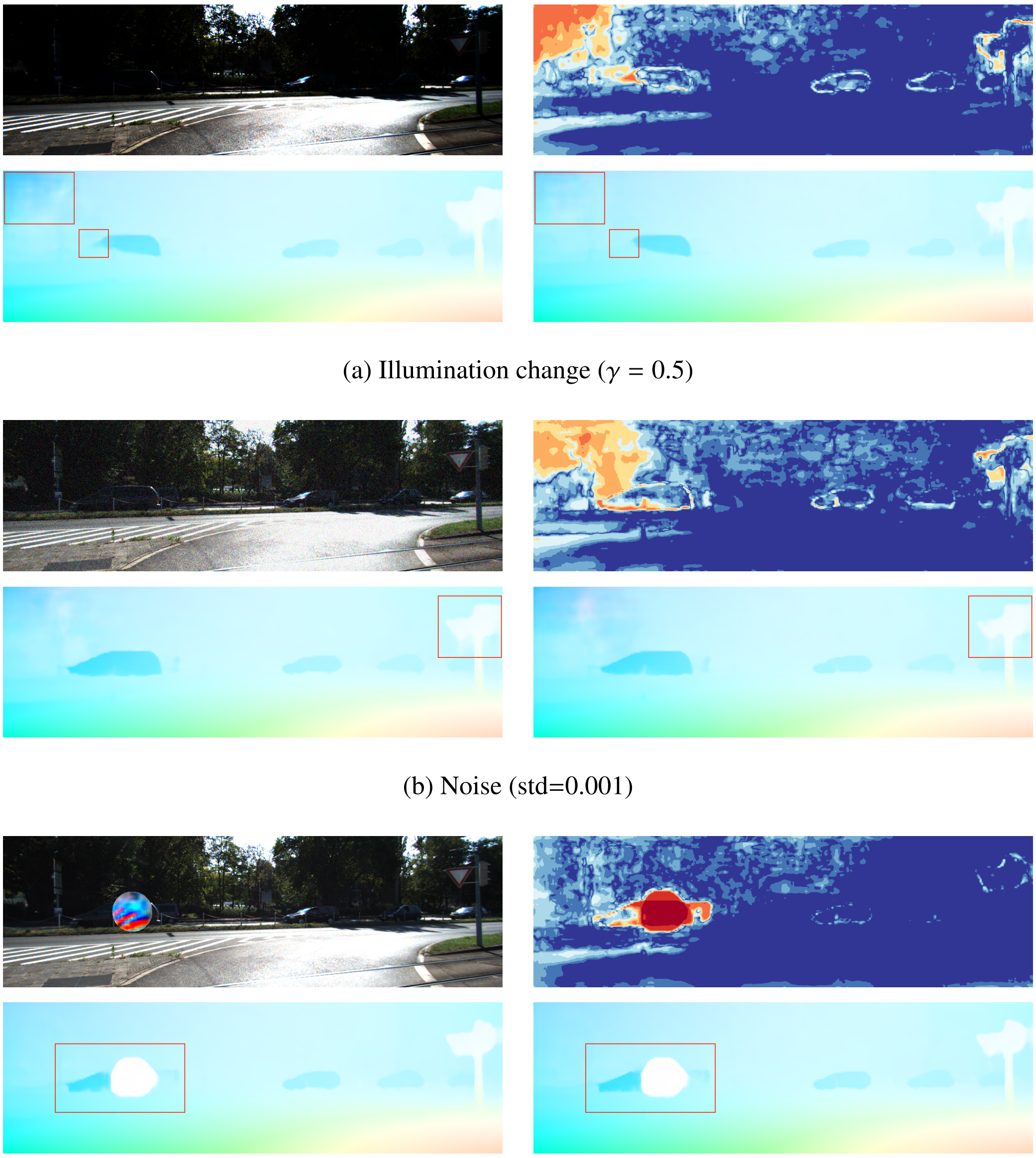
(c) Adversarial patch

Patch size	50	100	150	200
VCN	3.5/0.981	5.6/1.419	8.5/2.048	11.9/2.880
VCN+LCV	3.4/0.949	5.5/1.384	8.3/2.004	11.6/2.801

2.4.4 Robustness Analysis

To further understand the effect of the LCV module, we evaluate the flow estimation performance under three challenging cases, *i.e.,* 1) illumination changes: we adjust the illumination of the input frames by changing the value of γ, where $\gamma = 1.0$ is the original image, $\gamma < 1.0$ is for a darker image, and $\gamma > 1.0$ is for a brighter image. 2) adding noises: we adjust the standard deviation to control the noise magnitude. and 3) inserting adversarial patches: we borrow the universal adversarial patch [100] that can perform a black-box attack for all optical flow models, and insert patches of different sizes to the input frames.

We compare the VCN model and its variant equipped with LCV. Both two models are trained on the KITTI 2015 training set. For qualitative comparison, we perform the above three types of processing on 194 images with the flow groundtruth from the KITTI



(a) Illumination change ($\gamma = 0.5$)

(b) Noise (std=0.001)

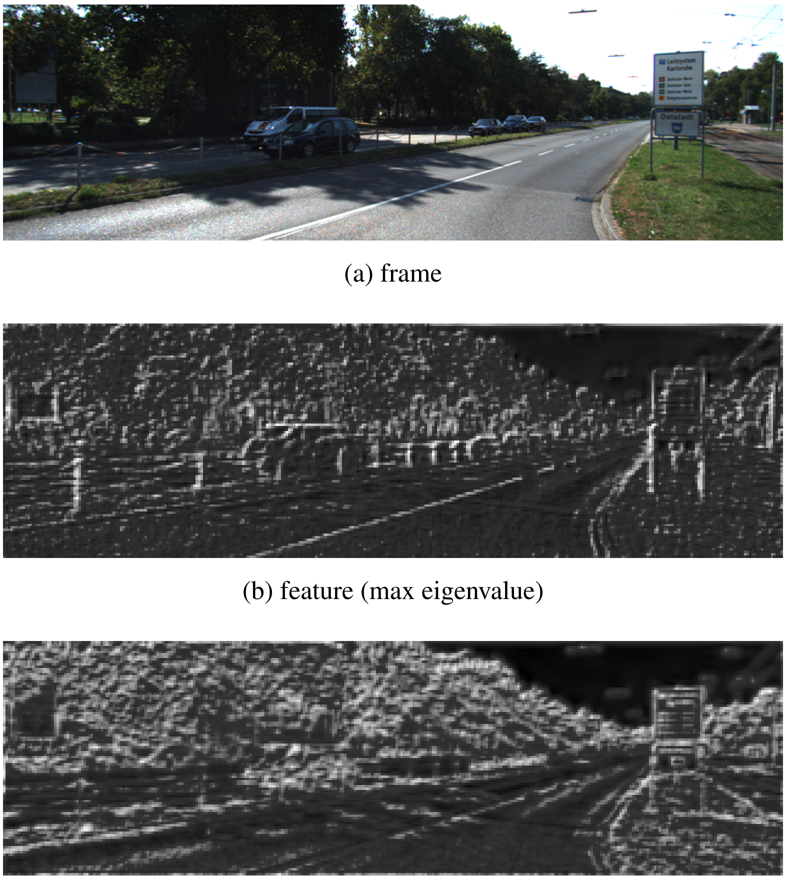
(c) Adversarial patch (radius=50)

Figure 2.6: Visual results of three challenging cases, *i.e.,* illumination change, noise, and adversarial patch. Top left: the first input frame; bottom left/right: flow by VCN / VCN+LCV; top right: flow difference between two methods.

2012 as our test set. As shown in Table 2.4(a), VCN+LCV consistently outperforms the VCN baseline in all three challenging cases. For better illustration, we visualize the effect on an image from KITTI 2015 test set as shown in Fig. 2.6. It can be seen that the LCV module can help stabilize the flow prediction around the background trees at the top left corner of the frame under the cases of dark illumination and random noise injection. In the third example, the outline of the car body near the patch circle is better preserved by our model. (See the difference map for details.)

2.4.5 Visualization of the Learned Features

We visualize the feature maps for different eigenvalues in Fig. 2.7. We find that boundaries of (moving) objects are salient in the feature map corresponding to the max eigenvalue while the min eigenvalue mainly corresponds to background information, which results in more discriminative cost volume and more accurate flow estimation.



(a) frame

(b) feature (max eigenvalue)

(c) feature (min eigenvalue)

Figure 2.7: The feature maps corresponding to the largest the smallest eigenvalues.

2.5 Proofs of Theorems

2.5.1 Proof of Theorem 1

The Theorem of Cayley representation was first given by Cayley in his paper [8]. However, the old paper is not available online. For convenience, we give a simple proof here.

Proof. First, we validate that the $\boldsymbol{P}$ is a orthogonal matrix. The condition that $\boldsymbol{S} \in \mathrm{SO}^*(n)$ ensures that $(\boldsymbol{I} + \boldsymbol{S})$ is invertible. Since $\boldsymbol{S}$ is skew-symmetric, so $\boldsymbol{S}^\top \boldsymbol{S} = -\boldsymbol{S}^2 = \boldsymbol{S}\boldsymbol{S}^\top$. Hence we have

$$\begin{aligned}
\boldsymbol{P}^\top \boldsymbol{P} &= (\boldsymbol{I} + \boldsymbol{S})^{-\top}(\boldsymbol{I} - \boldsymbol{S})^\top(\boldsymbol{I} - \boldsymbol{S})(\boldsymbol{I} + \boldsymbol{S})^{-1} && (2.15)\\
&= (\boldsymbol{I} + \boldsymbol{S}^\top)^{-1}(\boldsymbol{I} - \boldsymbol{S}^\top)(\boldsymbol{I} - \boldsymbol{S})(\boldsymbol{I} + \boldsymbol{S})^{-1} && (2.16)\\
&= (\boldsymbol{I} - \boldsymbol{S})^{-1}(\boldsymbol{I} - \boldsymbol{S}^\top - \boldsymbol{S} - \boldsymbol{S}^\top\boldsymbol{S})(\boldsymbol{I} + \boldsymbol{S})^{-1} && (2.17)\\
&= (\boldsymbol{I} - \boldsymbol{S})^{-1}(\boldsymbol{I} - \boldsymbol{S}^\top - \boldsymbol{S} - \boldsymbol{S}\boldsymbol{S}^\top)(\boldsymbol{I} + \boldsymbol{S})^{-1} && (2.18)\\
&= (\boldsymbol{I} - \boldsymbol{S})^{-1}(\boldsymbol{I} - \boldsymbol{S})(\boldsymbol{I} - \boldsymbol{S}^\top)(\boldsymbol{I} + \boldsymbol{S})^{-1} && (2.19)\\
&= (\boldsymbol{I} - \boldsymbol{S}^\top)(\boldsymbol{I} + \boldsymbol{S})^{-1} && (2.20)\\
&= (\boldsymbol{I} + \boldsymbol{S})(\boldsymbol{I} + \boldsymbol{S})^{-1} && (2.21)\\
&= \boldsymbol{I}. && (2.22)
\end{aligned}$$

Next, we need to show the uniqueness of the Cayley representation.

$$\begin{aligned}
&\boldsymbol{P} &&= (\boldsymbol{I} - \boldsymbol{S})(\boldsymbol{I} + \boldsymbol{S})^{-1} && (2.23)\\
\Longleftrightarrow\ &\boldsymbol{P}(\boldsymbol{I} + \boldsymbol{S}) &&= \boldsymbol{I} - \boldsymbol{S} && (2.24)\\
\Longleftrightarrow\ &\boldsymbol{P} + \boldsymbol{P}\boldsymbol{S} &&= \boldsymbol{I} - \boldsymbol{S} && (2.25)\\
\Longleftrightarrow\ &\boldsymbol{S} + \boldsymbol{P}\boldsymbol{S} &&= \boldsymbol{I} - \boldsymbol{P} && (2.26)\\
\Longleftrightarrow\ &(\boldsymbol{I} + \boldsymbol{P})\boldsymbol{S} &&= \boldsymbol{I} - \boldsymbol{P} && (2.27)\\
\Longleftrightarrow\ &\boldsymbol{S} &&= (\boldsymbol{I} + \boldsymbol{P})^{-1}(\boldsymbol{I} - \boldsymbol{P}) && (2.28)
\end{aligned}$$

Therefore, the skew-symmetric matrix $\boldsymbol{S}$ is uniquely represented by $\boldsymbol{P}$, which concludes the proof. □

2.5.2 Proof of Theorem 2

Before giving the proof, we would like to recall the definition of connectedness.

Definition 1 (Connectedness)**.** *A set of matrices $\mathcal{G}$ is said to be connected if for all $\boldsymbol{A}$ and $\boldsymbol{B}$ in $\mathcal{G}$, there exists a continuous path $\boldsymbol{A}(t)$, $0 \le t \le 1$, lying with $\boldsymbol{A}(0) = \boldsymbol{A}$ and $\boldsymbol{A}(1) = \boldsymbol{B}$.*

The above definition of connectedness is actually path connectedness in topology. Now we begin our proof of Theorem 2.

Proof. Since the identity matrix $\boldsymbol{I} \in \mathrm{SO}^*(n)$, it suffices to prove that for any $\boldsymbol{X} \in \mathrm{SO}^*(n)$, there exists a continuous path $\boldsymbol{A}(t)$, $0 \le t \le 1$, such that $\boldsymbol{A}(0) = \boldsymbol{I}$ and $\boldsymbol{A}(1) = \boldsymbol{X}$. For any $\boldsymbol{X} \in \mathrm{SO}^*(n)$, we have its spectral decomposition

$$\boldsymbol{X} = \boldsymbol{P}^\top \mathrm{diag}(K_1, \ldots, K_q, 1, \ldots, 1)\boldsymbol{P}, \tag{2.29}$$

where the $\boldsymbol{P} \in \mathrm{O}(n)$ and $0 \le q \le n/2$, and

$$K_\lambda = \begin{pmatrix} \cos(\theta_\lambda) & -\sin(\theta_\lambda) \\ \sin(\theta_\lambda) & \cos(\theta_\lambda) \end{pmatrix}, \quad \theta_\lambda \in [-\pi, \pi), \quad \lambda = 1, \ldots, q. \tag{2.30}$$

If we put

$$K_\lambda(t) = \begin{pmatrix} \cos(t\theta_\lambda) & -\sin(t\theta_\lambda) \\ \sin(t\theta_\lambda) & \cos(t\theta_\lambda) \end{pmatrix}, \tag{2.31}$$

then the path required is

$$\boldsymbol{A}(t) = \boldsymbol{P}^\top \mathrm{diag}(K_1(t), \ldots, K_q(t), 1, \ldots, 1)\boldsymbol{P}. \tag{2.32}$$

□

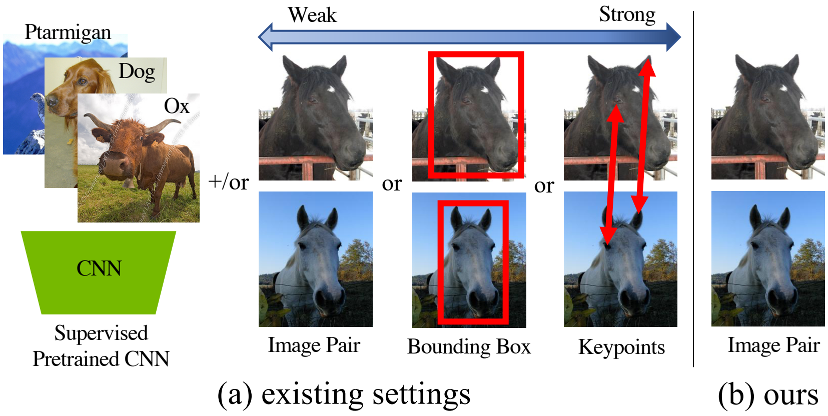
2.6 Conclusion

In this chapter, we introduce a learnable cost volume (LCV) module for optical flow estimation. The proposed LCV module generalizes the standard Euclidean inner product into an elliptical inner product with a symmetric and positive definite kernel matrix. To keep its symmetry and positive definiteness, we use the Cayley representation to re-parameterize the kernel matrix for end-to-end training. The proposed LCV is a lightweight module and can be easily plugged into any existing networks to replace the vanilla cost volume. Experimental results show that the proposed LCV module improves both the accuracy and the robustness of state-of-the-art optical flow models.

Chapter 3

Learning Contrastive Representation for Semantic Correspondence

Dense correspondence across semantically related images has been extensively studied but still faces two challenges: 1) large variations in appearance, scale, and pose exist even for objects from the same category, and 2) labeling pixel-level dense correspondences is labor-intensive and infeasible to scale. Most existing methods focus on designing various matching modules using fully-supervised ImageNet pretrained networks. On the other hand, while a variety of self-supervised approaches are proposed to explicitly measure image-level similarities, correspondence matching the pixel level remains under-explored. In this work, we propose a multi-level contrastive learning approach for semantic matching, which does not rely on any ImageNet pretrained model. We show that image-level contrastive learning is a key component to encourage the convolutional features to find correspondence between similar objects, while the performance can be further enhanced by regularizing cross-instance cycle consistency at intermediate feature levels. Experimental results on the PF-PASCAL, PF-WILLOW, and SPair-71k benchmark datasets demonstrate that our method performs favorably against the state-of-the-art approaches. The source code and trained models are available at `https://github.com/NVlabs/Contrastive-Correspondence`.



(a) existing settings

(b) ours

Figure 3.1: Visual comparison of existing and our settings in terms of the used supervision. Most existing works rely on supervised pretrained CNN and/or pairwise supervision. From weak to strong, the pairwise supervision includes image pair, bounding box, and keypoints. In contrast, our method only requires that image pairs belong to the same category for training.

3.1 Introduction

Semantic correspondence is one of the fundamental problems in computer vision with many applications in object recognition [23, 75], image editing [17], semantic segmentation [60], and scene parsing [152], to name a few. The goal is to establish dense correspondences across images containing the objects or scenes of the same category. For example, as shown in Fig. 3.1(a), such dense correspondence is established across two horse images, where the semantically similar keypoints such as eyes or ears of different horses are matched. However, this task is extremely challenging as different objects usually appear with distinctive appearances caused by variations in shapes, lighting, poses and scales. Classic approaches [5, 47, 60, 75, 114, 138] determine correspondence matching via hand-crafted features such as SIFT [79], DAISY [138] and HOG [16]. More recently, deep CNN based approaches [14, 34, 54, 57, 62, 93, 103, 104, 107, 151] have achieved significant improvements, by exploiting hierarchical image features that provide rich semantic features that are invariant to intra-class variations.

It is challenging to develop fully-supervised approaches for learning pixel-level matching as a large number of image pairs with detailed pixel correspondence anno-

Table 3.1: Supervisory signals used by evaluated methods. The last column indicates that the validation image pairs with keypoint annotations are used for model selection. Note that knowing keypoint correspondence between two images indicates that they form an image pair of the same category, and the bounding boxes of these objects can thus be extracted.

Methods	Supervised	Training			Validation
	Pretrained CNN	Image Pair	Bounding Box	Keypoints	Keypoints
DCTM [63]	✓			✓	✓
SC-Net [34]	✓			✓	✓
Weakalign [104]	✓	✓			✓
RTNs [61]	✓	✓			✓
NC-Net [105]	✓	✓			✓
DCC-Net [44]	✓	✓			✓
HPF [86]	✓				✓
DHPF [87]	✓	✓			✓
SF-Net [69]	✓		✓		✓
PARN [54]	✓		✓		✓
FCSS [62]	✓		✓		✓
SCOT [78]	✓				✓
ours		✓			

tations are required. To alleviate this issue, several methods exploit weakly-supervised information, e.g., object bounding boxes or foreground masks [54, 62, 152, 151, 69], for this task. Nevertheless, it still entails significant amount of labor to annotate these labels for a large scale dataset. As a trade-off between model performance and labeling effort, several methods [104, 61, 105, 44, 87] leverage image class labels as weak supervisory signals. For example, we can obtain image pairs from the same object or scene category, which provides weak supervision for learning the correspondence. However, existing methods predominantly rely on effective universal feature representations in the first place, which are often obtained by supervised ImageNet pretraining. Benefiting from large-scale labeled images, a supervised pretrained network can extract image-level discriminative features for semantic correspondence.

Recent advances [42, 9, 37, 29] in self-supervised representation learning exploit contrastive loss functions to construct image-level discriminative features from unlabeled image data. Their goal is to push object representations of various views of the same object closer while pulling those of different objects further apart by learning to perform an image instance discrimination task. The learned representations by these contrastive learning methods have achieved significant performance gains for numerous pretext tasks such as video object segmentation [97] and tracking [127]. However, learning a fine-grained representation at part- or pixel-level has not been well studied, especially for semantic correspondence without using ImageNet pretrained models. Some methods [129, 94] have generalized image-level contrastive learning to dense contrastive learning, but do not establish cross-instance correspondence. As shown in Fig. 3.1 and summarized in Table 3.1, existing approaches have used different levels of supervisory signals, while our model is learned based on weak supervision, i.e., image pairs of the same category, without resorting to fully-supervised pretrained ImageNet networks.

To learn a generalizable representation for semantic correspondence, we develop a multi-level contrastive representation learning method in this chapter. We show that by applying the contrastive learning framework [37] merely on the image level, the mid-level convolutional features can capture local correspondences between similar objects reasonably well. The results suggest that to learn good representations for high-level tasks (e.g., object recognition), lower-level features are enforced to learn correct correlations at a fine-grained level as well. We embed a pixel-level contrastive learning scheme into the same network to further obtain fine-grained feature representation learning by enforcing cross-instance cycle consistency regularization at intermediate feature levels. Given a pair of images with semantically similar objects, we track selected pixels from the source to the target images and then back to the source via an affinity matrix. We then enforce that those selected pixels map back to their original locations via the cross-instance cycle consistency constraint. Essentially, cycle consistency is equivalent to pixel-level contrastive learning, where the path of each pixel can be considered as either positive or negative depending on whether it forms a cycle [51]. To avoid trivial solutions, we use a self-attention module to localize the foreground object and apply a group of augmentations based on the computed attention map. The main contributions of this work can be summarized as follows:

- We pose a new weakly-supervised semantic matching problem by relaxing the strong dependency on supervised ImageNet pretrained models and removing the validation ground truth used for model selection.
- We develop a multi-level contrastive learning framework where image-level contrastive learning generates object-level discriminative representations and pixel-level contrastive learning further facilitates fine-grained representations at region- or pixel-level to improve dense semantic correspondence performance.
- We propose a cross-instance cycle consistency regularization to learn a discriminative local feature at the pixel-level without dense ground truth by leveraging different objects of the same category in different images instead of the same object in self-augmented images or video sequences.
- We demonstrate that the proposed model performs favorably against the state-of-the-art method on three datasets with comprehensive ablation studies on components of our framework.

3.2 Related Work

3.2.1 Semantic Correspondence

Numerous methods exploit hierarchical features from deep models pretrained on the ImageNet dataset to infer semantic correspondence. In these approaches, semantic matching is formulated as a geometric alignment task and addressed via the self-supervised learning framework where training image pairs and ground truth are synthesized based on in-plane transformations [54, 57, 61, 103, 104, 34, 107]. On the other hand, weak-supervision signals, such as image-level labels [104, 61, 105, 44, 87], bounding boxes [54, 62, 152, 151, 69], and keypoints [34] have also been used for semantic correspondence. In contrast, the proposed method does not require fully supervised ImageNet pretrained networks and only utilizes image-level supervision, i.e., image pairs of same category, to determine semantic correspondence from images.

In addition, numerous approaches freeze the ImageNet pretrained backbone model and determine the network structure of other modules or hyper-parameters by exploiting ground truth keypoint correspondences of a small validation set as supervisory signals [86, 78, 44, 12]. Min *et al.* [86] leverage a small number of relevant features selected from early to late layers of a convolutional neural network as well as beam search to construct hyperpixel layers, and use regularized Hough matching (RHM) to infer semantic correspondence efficiently. Liu *et al.* [78] pose semantic matching as an optimal transport problem and solve it using the Sinkhorn's algorithm [111]. These two methods require only supervision from keypoints of a small validation set during the beam search stage to select feature layers from a pretrained deep model (e.g., ResNet-50 and ResNet-101). We note that keypoint ground truth from a validation set provides robust guidance for model and hyper-parameter selection, a level of supervision that is stronger than either pixel-level training supervision or supervised ImageNet pretrained model weights. In contrast, our method does not require any keypoint correspondence supervision from a validation dataset as the proposed cross-instance cycle consistency regularization can be used for self-supervised learning.

3.2.2 Self-Supervised Representation Learning

Self-supervised representation learning methods can be classified into three categories: generative, inductive (predictive), and contrastive. Generative models, e.g., [122, 119, 120], maximize the likelihood of observed data based on various probabilistic formulations and representation, e.g., auto-encoders and convolutional neural networks. In contrast, inductive methods predict some known information from the data as there exist ground truths for comparison, which is typically carried out by applying various augmentation techniques, including image rotations prediction [28], spatial configuration of cropped patches [19], video colorization [147], video sequence order sorting [89], and jigsaw puzzle solving [92].

Recently, numerous contrastive learning methods, such as Deep InfoMax [42], SimCLR [9], MoCo [37] and BYOL [29], have shown that effective representation models can be constructed without supervision, with performance comparable to fully supervised ones. However, image-level contrastive learning may not produce optimal features for semantic correspondence, where the association is built between pixels. A few methods [129, 94, 51] propose to learn dense visual representations where the association is required between pixels, which are direct extensions of image-level instance contrastive learning methods (e.g., MoCo [37]). They compute contrastive losses between different views of the *same* instance, e.g., either via self-augmented images [129, 94], or from videos of the same instance [51]. However, their learned representations only captures the instance-level information, and thus cannot be used to establish the pixel-level cross-instance semantic correspondence. In contrast, we propose a multi-level cross-instance pixel contrastive learning method by leveraging *different* instances via a cyclic framework to infer semantic correspondence.

Kang *et al.* [58] leverage the pixel-level cycle association of source and target pixel pairs across two different domains for contrastive representation learning. However, it requires knowing the pixel-level categories in advance to construct positive/negative samples for contrastive learning. In contrast, the proposed method does not entail any pixel-level supervisory signals to form positive/negative pixel pairs. On the other hand, Xie *et al.* [135] propose a joint pixel-level and instance-level contrastive learning framework. It differs from our approach in constructing positive and negative samples. The pixel-level contrastive learning in [135] is carried out through pixels within the same instance, i.e., the corresponding pixels in two views of the same image are considered as positive pairs. Nevertheless, the proposed model establishes pixel-level contrastive learning across different instances. Extensive experimental results show that the proposed cross-instance pixel-level contrastively learned representation can better handle challenges caused by large appearance and pose distinctions across different instances.

3.2.3 Temporal Correspondence

Our work is related to temporal correspondence learning, for which a number of self-supervised approaches [124, 95, 71, 128, 51] have been explored. Liu *et al.* [71] propose to learn temporal correspondence by joining region-level localization and pixel-level matching through a shared inter-frame affinity matrix. In [51], the correspondence learning problem is cast as link prediction in a space-time graph constructed from a video and long-range cycle consistency is exploited to learn temporal correspondence.

Unlike these methods, we target a more challenging task for establishing correspondence across semantically similar instances with significant appearance and pose distinctions.

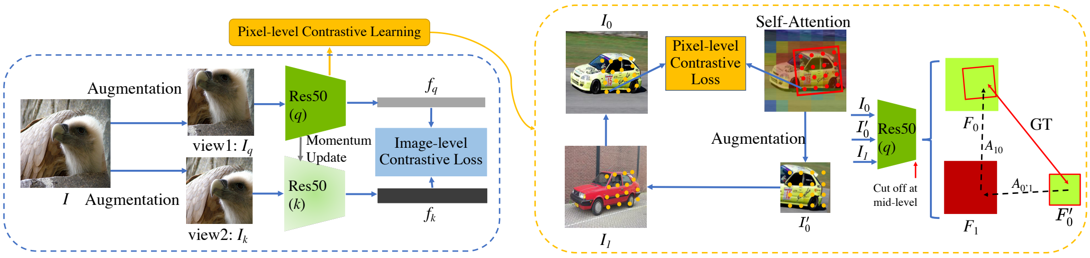
3.2.4 Cycle Consistency

Cycle consistency has been widely used as a constraint in numerous vision tasks. For example, in the context of image-to-image translation [154], or face attributes editing [150, 133, 134], exploiting cycle consistency enables learning the mapping between different image domains from unpaired data. In the context of optical flow estimation, computing the forward and backward consistency [82, 76] can be used to effectively infer occluding pixels for learning optical flow.

For learning correspondence, the cycle consistency constraint is formulated in various ways. Zhou *et al.* [151] construct a cross-instance loop between real and synthetic images and establish cross-instance correspondence through 3D CAD rendering. SC-Net [34] establishes a differentiable flow field by computing feature similarities while considering background clutter and proposes flow consistency loss between forward and backward flow fields. Zhou *et al.* [153] propose to optimize joint matching of multiple images via rank minimization by translating cycle consistency into positive semi-definiteness and low-rankness constraints. On the other hand, Chen *et al.* [12] exploit forward-backward consistency and transitivity consistency constraints to enforce geometrically plausible predictions. In contrast, our cycle consistency regularization is established via self augmentations to learn finer-grained pixel-level representations.

3.3 Proposed Method

In this section, we introduce the proposed contrastive representation learning model for semantic correspondence. As shown in Fig. 3.2, the overall framework consists of image-level and pixel-level contrastive learning modules. We briefly describe the image-level contrastive learning schemes in Section 3.3.1, introduce the proposed pixel-level contrastive learning method in Section 3.3.2 and our implementation details in Section 3.3.3.



(a) Image-level Contrastive Learning (random images from ImageNet)

(b) Pixel-level Contrastive Learning (images pairs, e.g., from PF-PASCAL)

Figure 3.2: Overview of our framework. The dark green trapezoid denotes the backbone network q shared by (a) image-level and (b) pixel-level contrastive learning. In (a), we use random images from ImageNet as the inputs, and the feature from its last layer for image-level contrastive learning, where the image-level contrastive loss is defined as (3.1). In (b), we use image pairs of the same category (e.g., from PF-PASCAL) as inputs and extract their mid-level features for pixel-level contrastive learning, where the pixel-level contrastive loss is defined as (3.5).

3.3.1 Image-level Contrastive Learning

Image-level contrastive learning [37, 9, 29] aims to extract object-level discriminative features in a self-supervised manner. It learns representations from object-centric images by minimizing the distance between two different views of an object generated through different augmentation methods. In addition, numerous methods [37, 9] use negative samples from different images and maximize the distance between positive and negative observations. In this work, we exploit negative examples in a way similar to the MoCo method [37]. Image-level contrastive learning can be considered as a dictionary look-up task. We use a dynamic queue of size K to store a set of encoded keys $\{f_1, f_2, \ldots\}$, among which a single positive key f_k matches with f_q. For a query f_q, its positive key f_k encodes a different view of the same image, while the negative keys encode the views of different images. Thus, the image-level contrastive loss is defined as follow:

$$L_q = -\log \frac{\exp(f_q \cdot f_k/\tau)}{\exp(f_q \cdot f_k/\tau) + \sum_{i \neq k} \exp(f_q \cdot f_i/\tau)}, \tag{3.1}$$

where τ denotes a temperature hyper-parameter. Both the query network q and the key network k share the same architecture, but q is updated based on the image-level contrastive loss using back-propagation while k is updated with a momentum function:

$$\theta_k \leftarrow m\theta_k + (1 - m)\theta_q, \tag{3.2}$$

where θ_q and θ_k are parameters of q and k, and $m \in [0, 1)$ denotes the momentum coefficient.

For image-level contrastive learning, each image is regarded as a unique category. Therefore the function for image-level contrastive learning approaches (3.1) is essentially equivalent to a $(K + 1)$-way classification task, where K is the size of the queue. It facilitates learning image-level discriminative representations without using any image labels. We show in our experiments (see Table 3.3) that even without additional objective functions, the learned image-level representation models can perform semantic correspondence matching well (see Fig. 3.3).

3.3.2 Pixel-level Contrastive Learning

To learn fine-grained representation models, we propose a pixel-level contrastive learning method explicitly for semantic correspondence. As shown in Fig. 3.2(b), a pair of images I_0 and I_1 containing different objects of the same category are provided as the inputs. We then obtain an attention map highlighting the foreground object region via the self-attention module. Based on the attention map, we obtain I'_0 by applying a series of spatial data augmentations to I_0, including random horizontal flipping, rotation, and cropping. We use the same encoder network q, shared with the image-level module, to extract features for I_0, I_1, and I'_0, which are denoted by F_0, F_1 and F'_0, respectively, as shown in Fig. 3.2(b)). We extract feature maps from the middle layers of ResNet50 instead of the feature vector from the last layer to learn the pixel-level representation. We introduce each sub-module in the following sections.

Self-attention module. We introduce a self-attention module to avoid including pixels from background regions. For the given source image I_0, we extract the feature $f_0 \in \mathbb{R}^C$ after the last layer and the feature $K_0 \in \mathbb{R}^{C \times H \times W}$ before the global average pooling layer, where H and W denote the size of the feature map. As the feature map K_0 keeps the spatial location information of pixels in I_0, we compute cosine similarity between normalized f_0 and all feature vectors in the normalized attention map K_0. The self-attention module is used to exclude background pixels when applying a series of augmentation to I_0 to obtain I'_0. The random image regions are cropped near the pixels using the largest cosine similarity. As such, the background pixels are less likely to be contained in the augmented image I'_0, thus eliminating the number of pixel paths starting from the background for more effective cross-instance cycle consistency.

Associating pixels via affinity. Given a pair of images I_0 and I_1 and their mid-level features $F_0 \in \mathbb{R}^{C \times H_0 W_0}$ and $F_1 \in \mathbb{R}^{C \times H_1 W_1}$, we use the correlation matrix $R_{01} \in \mathbb{R}^{H_0 W_0 \times H_1 W_1}$ to represent the pixel-level similarity as:

$$R_{01} = F_0^\top F_1. \tag{3.3}$$

To ensure a one-to-one mapping, the correlation matrix needs to be sparse. However, it is challenging to model a sparse matrix in a deep neural network. Therefore, we relax this constraint and encourage the correlation matrix to be sparse by normalizing each row with the softmax function. As such, the similarity score distribution can be peaky, and only a few pixels with high similarity in the source image are matched to each point in the target image. The affinity matrix is defined by:

$$A_{01} = \mathrm{softmax}(R_{01}/t), \tag{3.4}$$

where t is the temperature hyperparameter controlling how peaky the normalized distribution is. The affinity matrix enjoys several good properties: 1) The summation over each row is unity since softmax is applied to the row dimension. 2) The multiplication of two affinity matrices results in an affinity matrix. 3) The affinity matrix can be used to trace the corresponding pixel locations of feature map F_0 in the target feature map F_1, defined by $P_{01} = A_{01}G_1$, where $G_1 \in \mathbb{R}^{H_1 W_1 \times 2}$ is a vectorized pixel location map (i.e., each element denotes its horizontal and vertical positions).

Pixel-level guidance. We carry out the proposed pixel-level contrastive learning by formulating the correspondence as a graph, where the nodes are image pixels and the edges are weighted by the similarities between their features in the latent space. Starting from I'_0 to I_1 and then back to I_0, we track corresponding pixels by computing two affinity matrices $A_{0'1}$ and A_{10} following (3.4) using their mid-layer features. We enforce cross-instance cycle consistency by requiring pixels from I'_0 to be mapped back to where they are located in I_0, through a different image I_1. The affinity matrix along the path of cycle can be simply obtained through multiplication, $\bar{A}_{0'0} = A_{0'1}A_{10} \in \mathbb{R}^{H'_0W'_0 \times H_0W_0}$, where H and W denote the feature size. Each element in the cycle affinity matrix $\bar{A}_{0'0}$ depicts a pixel path from I'_0 to I_0 passing through I_1. We predict corresponding pixel locations of the patch image I'_0 in the source image I_0 by $P = \bar{A}_{0'0}G_0$. As the ground truth correspondence $\hat{P}$ between I'_0 and I_0 is known, the pixel-level contrastive loss is defined as:

$$L_p = ||P - \hat{P}||_2. \tag{3.5}$$

We note that (3.5) is a variant of the contrastive objective w.r.t. pixels from a pair of images. Unlike the image-level loss (3.1) that explicitly pushes and pulls on the positive and negative pairs, the cycle loss matches a group of "starting" and "ending" pixels. Specifically, all pixels sampled in the walking path are considered positive, while the other pairs are negative. The feature representation is learned to be pixel-wisely discriminative when the affinity matrix is enforced to be "peaky" in each row.

Information entropy regularization. Information entropy loss [87] can also be used as a regularization term to learn affinity by encouraging more distinctive correspondences. We empirically find that training the network by the pixel-level contrastive loss with the information entropy loss can further improve the performance of semantic correspondence. With the correlation matrix $R \in \mathbb{R}^{H_0W_0 \times H_1W_1}$ as defined in (3.3), we compute the correlation entropy as:

$$H(R) = -\frac{1}{H_0W_0} \sum_{i=1}^{H_0W_0} \sum_{j=1}^{H_1W_1} \phi(R)_{ij} \log \phi(R)_{ij}, \tag{3.6}$$

where $\phi(\cdot)$ denotes row-wise ℓ_1 normalization, and $\phi(R)_{ij}$ denotes the (i, j)-th elements of $\phi(R)$. As lower correlation entropy indicates more distinctive correspondence between two images, we encourage low entropy for the image pair I_0 and I_1 by using the following information entropy loss:

$$L_r = H(R_{01}) + H(R_{10}), \tag{3.7}$$

where R_{01} and R_{10} are correlation matrices between the source and target images. We use correlation matrices R instead of affinity matrices A to compute the information entropy loss. The reason is that each row in $A = \text{softmax}(R/t)$ has been normalized with summation of each row to 1, and thus all entries would have been normalized twice if we use $\phi(A)$.

3.3.3 Implementation Details

Pretraining and training. We use ResNet50 [38] as the backbone network for feature extraction. For image-level contrastive learning, we use the feature from the last layer of ResNet50, while the feature from an intermediate layer (e.g., the feature after 13th Res-block) for pixel-level contrastive learning. Following the same hyperparameter settings of MoCo [37], we pretrain the backbone network using image-level contrastive learning on the ImageNet dataset for initialization. Then we train the backbone network with a combination of image-level and pixel-level contrastive losses along with the information entropy regularization,

$$L = \lambda_p L_p + \lambda_q L_q + \lambda_r L_r, \tag{3.8}$$

where λ_p, λ_q, and λ_r are weight coefficients for the pixel-level contrastive loss (3.5), image-level contrastive loss (3.1) and the information entropy regularization (3.7), respectively. Note that for image-level contrastive learning, we do not need a 1000-class labeled dataset, but just random images. Neither do we need that for pixel-level contrastive learning. In contrast, only image pairs from the same category are required for pixel-level contrastive learning. We empirically set $\lambda_p = 0.0005$, $\lambda_q = 1$, $\lambda_r = 0.001$, and the temperature $t = 0.0007$ in (3.4).

Validation: beam search without ground truth correspondence. To utilize the learned representation model for semantic correspondence, we need to choose multi-layer features for testing, which is also called *hyperpixel* construction [86]. Existing methods [86, 78] use the beam search algorithm to find the optimal subset of deep convolutional layers according to performance on the validation split. However, the existing beam search method requires accessing the ground truth correspondence of the validation set. To relax the dependency on validation annotations, we perform beam search over all convolutional layers of a given deep model by using the proposed pixel-level contrastive loss as the performance indicator. A lower pixel-level contrastive loss means a better layer combination. The validation process is only used for hyperpixel selection and not for other hyper-parameter selection.

Matching process. Given a pair of images, we extract their features and compute the affinity matrix based on the selected hyperpixels. Similar to the steps in SCOT [78], we first perform the optimal transport (OT) to relax the affinity matrix and obtain the transport matrix between two images. By viewing matching problem as an optimal transport (OT) problem, we perform image matching at a global perspective compared to matching pixels independently. The OT problem can be solved using Sinkhorn's algorithm [111]. Consequently, the many-to-one matching issue can be alleviated. Then we employ the regularized Hough matching (RHM) [86] as the post-processing step to obtain the voting matrix and determine final keypoint correspondences. The RHM method further improves the matching accuracy by enforcing geometric consistency by reweighting the matching score in the Hough space. We carefully analyse the effectiveness of these post-processing steps and the results can be found in Table 3.4 and Fig. 3.3.

3.4 Experiments and Analysis

We propose a new benchmark setting specifically for learning semantic correspondence without using any supervised ImageNet pretrained network or validation ground truth. For all the evaluated methods, we use the ResNet50 model [38] as the backbone network for feature extraction, in a way similar to the self-supervised pretraining scheme of [37]. We also use a unified standard for the validation step by performing model selection without resorting to using the ground truth keypoint correspondence of the validation image pairs.

In Section 3.4.1, we first describe the datasets and evaluation metrics. Then, we present results as per the new benchmark setting by comparing our method with state-of-the-art methods in Section 3.4.2. In Section 3.4.3, we conduct comprehensive ablation studies to analyze different components and variations of our model.

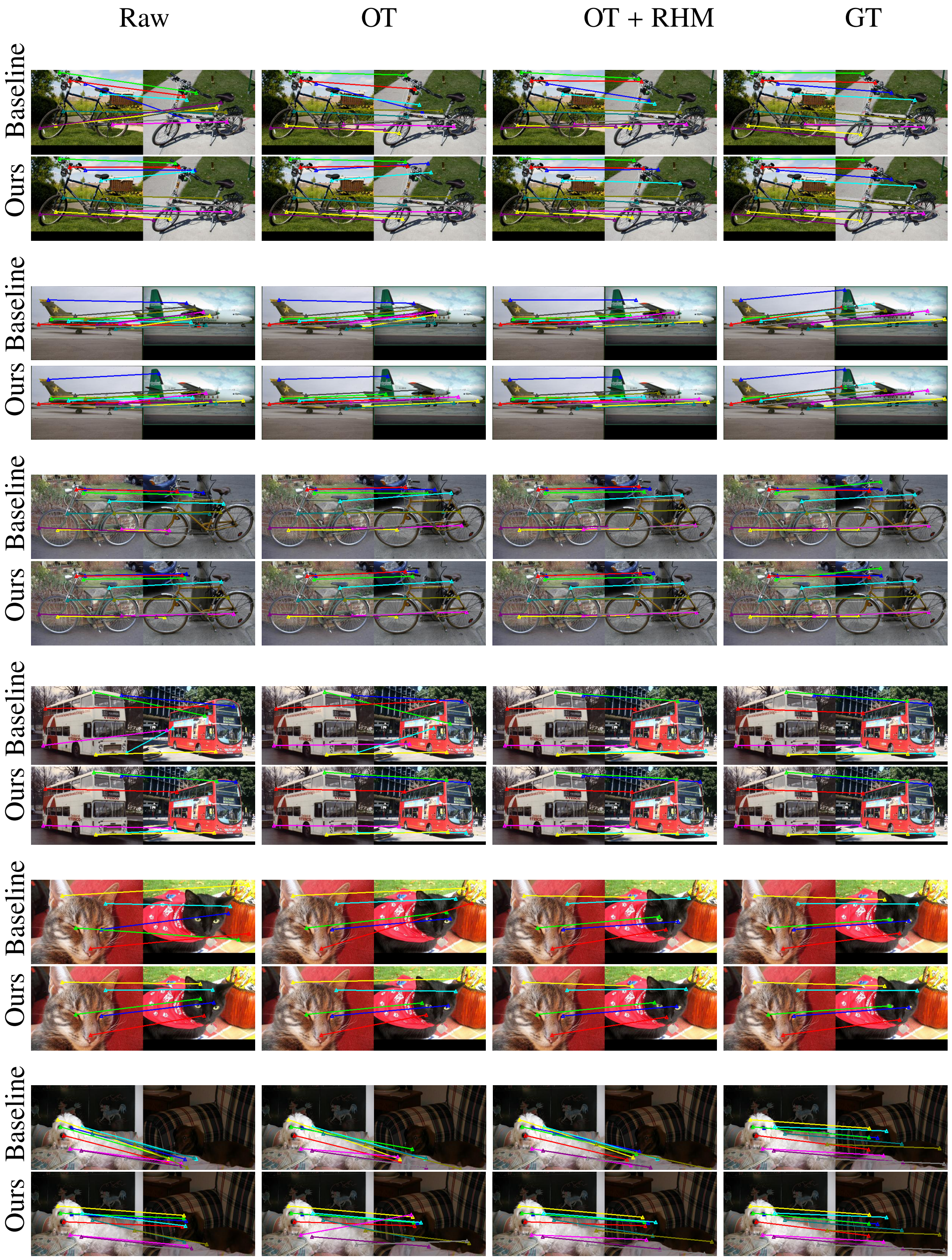


Figure 3.3: Visualization of the baseline and our method on the PF-PASCAL dataset. The first three columns show the correspondence predictions made based on the affinity matrix (Raw), transport matrix (after OT) and voting matrix (after OT + RHM). The last column is the ground truth correspondence. Different colors indicate different keypoint matches.

3.4.1 Datasets and Evaluation Metrics

We evaluate the proposed method on three benchmark datasets: PF-PASCAL [33], PF-WILLOW [32] and SPair-71k [88]. The PF-PASCAL dataset contains 1,351 image pairs from 20 object categories of the PASCAL VOC [24] database, and the PF-WILLOW dataset contains 900 image pairs of 4 object categories. The SPair-71k dataset is a more challenging large-scale dataset consisting of keypoint-annotated 70,958 image pairs from 18 categories with diverse view-point and scale variations. We carry out pixel-level contrastive learning on one of these three datasets and image-level contrastive learning method on the ImageNet dataset [18].

The PF-PASCAL dataset consists of 1300 image pairs with keypoint annotations of 20 object classes. Each pair of images in the PF-PASCAL dataset share the same set of non-occluded keypoints. We divide the dataset into 700 training pairs, 300 validation pairs, and 300 testing pairs following [87, 78]. The image pairs for training/validation/testing are distributed proportionally to the number of image pairs of each object class.

For quantitative evaluation, we adopt the widely-used percentage of correct keypoints (PCK) metric, which counts the number of correctly predicted keypoints given a fixed threshold. Given a predicted keypoint k_{pr} and the ground truth keypoint k_{gt}, the prediction is considered correct if:

$$d(k_{pr}, k_{gt}) \leq \alpha_\tau \cdot \max(w_\tau, h_\tau), \tag{3.9}$$

where $d(\cdot, \cdot)$ is the Euclidean distance, w_τ and h_τ are the width and height of either an entire image or object bounding box, i.e., $\tau \in \{\text{img}, \text{bbox}\}$, and α is a fixed threshold. We compute the final PCK by averaging the results from all testing image pairs.

3.4.2 Evaluation against the State-of-the-art Methods

Due to the full supervision and richer features of deeper neural networks (e.g., ResNet101), models that are pretrained with the large-scale labeled ImageNet have stronger representation strength. For fair comparisons, we *re-implement all the evaluated methods* by initializing their models with the same MoCo [37] pretrained ResNet50 backbone network. We emphasize that the backbone networks weights of NC-Net [105],

Table 3.2: Evaluation results on the PF-PASCAL and PF-WILLOW datasets. We compare our method against others with three different thresholds. Based on whether we add the entropy loss Eq. (3.7) or not, there are two variants of method, denoted by w/o and w/ entropy in the table. Numbers in bold indicate the best performance.

Methods	PF-PASCAL (α_{img})			PF-WILLOW (α_{bbox})		
	PCK@0.05	PCK@0.10	PCK@0.15	PCK@0.05	PCK@0.10	PCK@0.15
Weakalign [104]	30.83	60.90	76.49	27.98	56.29	72.42
NC-Net [105]	41.34	62.89	72.14	–	–	–
DCC-Net [44]	45.28	72.26	82.43	37.29	65.01	78.10
DHPF [87]	45.64	73.96	85.69	43.09	69.01	82.14
SCOT [78]	44.30	71.20	83.80	37.20	62.40	75.90
Ours(w/o entropy)	**51.00**	**77.10**	**88.50**	40.10	66.40	80.30
Ours(w/ entropy)	50.70	76.10	85.80	**43.10**	**69.80**	**82.40**

DCC-Net [44] and DHPF [87] are all frozen during training as stated in the experimental sections in these papers. The SCOT method [78] does not involve any training. Instead, it adopts a fixed backbone and optimizes on top of the extracted features. Since all the methods are built on top of a fixed pretrained network, it is fair to evaluate all methods by replacing their backbones with the same MoCo pretrained ResNet50 model. Replacing these models with purely randomly initialized weights will degrade the performance. Note that we specifically evaluate the methods that utilized the same level of weak supervision as ours, i.e., a group of matchable image pairs.

As aforementioned, the ground truth annotations of the validation set are utilized by numerous approaches, e.g., Weakalign [104], NC-Net [105], DCC-Net [44], DHPF [87], to determine the optimal hyper-parameters (neighborhood numbers, kernel size, etc.). For these method, we adopt the hyper-parameters in their original implementation in our re-implementations, because other randomly selected hyper-parameters would lead to lower performance. In addition, we utilize the re-implemented SCOT [78] method as our baseline model, which shares a similar matching procedure, i.e., OT and RHM, but with features extracted from a fixed backbone network pretrained with the image-level contrastive loss. For our implementation of SCOT, we remove the use of the validation set for beam search and replace it with the same strategy as proposed in Section 3.3.3.

Table 3.2 shows improvements introduced by our proposed multi-level contrastive learning method. We also visualize the results of the re-implemented SCOT baseline model and our method in Fig. 3.3. Our method generates more accurate and consistent matches than the baseline model (see more discussion of the reasons accounting for the improvement in the ablation study below). The re-implemented SCOT baseline model shows that the intermediate features of the backbone network pretrained using image-level contrastive learning can capture the semantic correspondences well. Furthermore, by embedding pixel-level contrastive learning into the network, the proposed model can adapt to the variations between different images at the pixel level, thereby generating more fine-grained matching results.

3.4.3 Ablation Studies

In this section, we analyze numerous components of the proposed method, including the image-level and pixel-level contrastive losses, beam search, OT and RHM, information entropy regularization, self-attention module, and the training layers. As each component plays an essential role in the model's performance, we conduct comprehensive ablation studies and discuss the contribution of each module.

Variants of image- and pixel-level contrastive learning. We analyze the proposed pixel-level contrastive learning with comparisons to other possible variants on the PF-PASCAL dataset, as summarized below and in Table 3.3. For simplicity, we use IC and PC as the abbreviations of image-level and pixel-level contrastive learning variants:

1. IC: We re-implement SCOT as our baseline model, where the backbone is pre-trained via image-level contrastive learning, as aforementioned in Section 3.4.2.

2. IC (finetune): To verify whether the performance gain above the baseline model comes from the image pairs in the PF-PASCAL dataset or not, we finetune the baseline model via the image-level contrastive loss on both Image-Net and PF-PASCAL datasets.

3. IC + PC (selfcycle): In addition to image-level contrastive learning, we conduct pixel-level contrastive learning on self-augmented image pairs from ImageNet.

4. IC + PC (align): We apply image- and pixel-level contrastive learning on both ImageNet and PF-PASCAL datasets, where the pixel-level contrastive loss of ImageNet is calculated on the self-augmented image pairs.

5. IC (fix) + PC: To verify the effectiveness of multi-level contrastive learning, we fix the backbone pretrained model via image-level contrastive learning while training a side convolutional net branch combining features from multiple layers as input via the pixel-level contrastive loss. The integrated feature following these additional layers is directly used for inference without applying beam search to the trained model.

6. IC + PC (ours): We jointly conduct image-level contrastive learning (on ImageNet) and pixel-level contrastive learning (on PF-PASCAL).

Table 3.3: Ablation study of the image- and pixel-level contrastive losses. All the numbers are evaluated at PCK@0.05 on the PF-PASCAL dataset. IC and PC are abbreviations of image-level contrastive learning and pixel-level contrastive learning. IN and PF stand for the ImageNet and PF-PASCAL datasets. Please refer to the ablation studies described in Section 3.4.3 for details.

Models	IC		PC		PCK
	IN	PF	IN	PF	@0.05
IC	✓				44.3
IC (finetune)	✓	✓			43.7
IC + PC (selfcycle)	✓		✓		38.1
IC + PC (align)	✓	✓	✓	✓	34.6
IC (fix) + PC				✓	39.5
IC + PC (Ours)	✓			✓	**51.0**

As shown in Table 3.3, the proposed IC + PC model outperforms the IC baseline method by a large margin (51.0% v.s. 44.3%). Either the effectiveness of pixel-level contrastive learning or using image pairs from the same category accounts for the performance gain. To better understand these effects, we conduct the IC (finetune) experiment.

As it performs almost as well as the IC variant (43.7% v.s. 44.3%), the performance gain of the IC + PC model does not result from the weak supervision provided by the PF-PASCAL dataset. In other words, the proposed pixel-level contrastive learning method helps achieve the performance gain.

Results of IC + PC (selfcycle) and IC + PC (align) show that using augmented image pairs from Image-Net for pixel-level contrastive learning negatively affects performance (38.1% and 34.6% v.s. 44.3%). The underlying reason is that different views of the same object may lead to trivial solutions as such augmented image pairs cannot provide views of different objects with a large variation. The result of IC (fix) + PC model is worse than that of the IC variant (39.5% v.s. 44.3%), which shows that model performance degrades without leveraging the image-level contrastive loss for updating the model weights. It also reflects that the image-level contrastive learning scheme is the cornerstone of the pixel-level contrastive learning scheme. The representations learned via the image-level contrastive learning scheme lead to coarse correspondences, and they can be refined via the pixel-level contrastive learning scheme, leading to more accurate correspondences.

Beam search, OT, and RHM. To validate the effectiveness of adopting the pixel-level contrastive loss as the objective in beam search (Section 3.3.3), we compare the results by performing beam search with and without using the ground truth annotations in the validation set. As shown in Table 3.4, beam search using only the pixel-level contrastive loss performs effectively in our method, as it gives nearly the same hyperpixel selection as the standard beam search algorithm (51.0% vs. 53.4%). This also reveals that the proposed pixel-level contrastive loss, as an unsupervised surrogate loss, facilitates learning effective fine-grained feature representations for cross-instance matching. In addition, we also study the effectiveness of OT and RHM. The results in Table 3.4 show that both OT and RHM effectively facilitate the matching process. More interestingly, even without using these two post-processing steps, our method achieves significant performance gain compared to the baseline approach (41.0% vs. 18.8% w/o GT). Fig. 3.3 shows some semantic correspondences by the evaluated methods. We note that OT and RHM regularize the correspondence by avoiding many-to-one mapping and by making all keypoints' correspondences geometrically consistent. Even without the

Table 3.4: Ablation study of beam search, optimal transport (OT), and regularized Hough matching (RHM). The w/o GT in the second column denotes the proposed beam search by using the pixel-level contrastive loss as the indicator. The numbers in the third column indicate the res-block IDs used for hyperpixel construction. The columns of OT and RHM denote whether they were used in the testing phase. The numbers in the last column are evaluated at PCK@0.05 on the PF-PASCAL dataset.

	Beam Search	Hyperpixel	OT	RHM	PCK @0.05
Baseline	w/o GT	(1,2,4,13,14)	✓	✓	44.3
	w/o GT	(1,2,4,13,14)	✓		20.9
	w/o GT	(1,2,4,13,14)			18.8
	w/ GT	(3,11,12,13,15)	✓	✓	52.4
	w/ GT	(3,11,12,13,15)	✓		39.1
	w/ GT	(3,11,12,13,15)			34.3
Ours	w/o GT	(2,12,13,15)	✓	✓	51.0
	w/o GT	(2,12,13,15)	✓		45.0
	w/o GT	(2,12,13,15)			41.0
	w/ GT	(2,11,12,15)	✓	✓	53.4
	w/ GT	(2,11,12,15)	✓		46.7
	w/ GT	(2,11,12,15)			41.3

post-processing steps (Fig. 3.3, Raw), our method can determine more geometrically consistent alignments than those by SCOT. These results demonstrate that the proposed pixel-level contrastive loss is effective in finding geometrically consistent matching by regularizing the learned representation.

Information entropy regularization. We validate the effectiveness of the information entropy loss (3.7). As shown in Table 3.2, the information entropy regularization effectively improves the performance on the PF-WILLOW dataset. However, it does not have the same effect on the PF-PASCAL dataset. We note that our method also significantly outperforms Weakalign [104] (Table 3.2), which solely utilizes the soft-inlier loss (a variant of entropy formulation).

Self-attention module. We evaluate the effectiveness of the self-attention module used for augmentation in the pixel-level contrastive learning method. As shown in Table 3.5, the self-attention module improves the matching performance of our model. Without employing the self-attention module, background pixels may be included in the cropped patch image I'_0. However, there is no semantic correspondence for those background pixels in the target image I_1. Taking the images in Fig. 3.2(b) as an example, the pixels in the grass regions of I_0 should not be matched to those in the wall regions of I_1 since they do not belong to the same category. The self-attention module can help remove the false positive samples in the background to achieve better results. More visualization of the attention map can be found in Fig. 3.4.

Table 3.5: Ablation study of the self-attention module on PF-PASCAL. We evaluate the performance at PCK@0.05.

	Self-Attention	PCK@0.05
Ours	✓	51.0
Ours (w/o attention)		49.9

Different layers of feature for training. We explore the effectiveness of features from different layers for pixel-level contrastive learning. For example, the ResNet50 model contains 16 res-blocks. We classify features before the 7th block as low-level features, those after the 13th block as high-level features, and the rest as middle-level features. Since features from shallow layers usually entail low-level vision clues, e.g., color and texture, and high-level layer features are invariant to different local regions (i.e., via image-contrastive loss), we only evaluate the mid-level features. Table 3.6 shows that the features from layer 7 to 13 help achieve similar performance while the 10th layer yields the best features. Even so, we utilize the features following 13th layer for pixel-level contrastive learning in the other experiments to fairly compare with other state-of-the-art methods.

Table 3.6: Evaluation results of using features from different layers on PF-PASCAL. The first column denotes the residual block ID where the feature for pixel-level contrastive learning is extracted. The second column represents the hyperpixel used for testing our beam search algorithm.

Block ID	Hyperpixel	PCK@0.05
13	(2,12,13,15)	51.0
12	(2,6,13,15)	49.1
10	(2,6,13,15)	52.0
9	(2,6,13,15)	51.3
7	(2,6,13,15)	49.1

Differentiable OT, RHM, and concentration loss [71]. We observe that both OT and RHM effectively improve the model performance when used as post-processing steps in the testing pipeline. Motivated by this, we analyze whether they can improve the performance in our model when including them in the training phase. We implement both OT and RHM as differentiable layers without trainable parameters. The function of the differentiable OT layer is equivalent to the Hungarian algorithm [90], classically used for bipartite matching. We apply one iteration of the Sinkhorn's algorithm to the affinity matrix and obtain the transport matrix. Differentiable RHM aims to achieve geometrically consistent matching by re-weighting the matching scores and outputs the voting matrix. Here we insert the differentiable OT and RHM layers into the training pipeline and compute the pixel-level contrastive loss on either transport matrix (after OT) or voting matrix (after OT and RHM) instead of the affinity matrix. In addition, we adopt the concentration loss [71] to encourage more concentrated keypoints predictions for the patch image.

By adopting differentiable OT and/or RHM layers, we observe an unstable training process. The results also reveal that the differentiable OT and RHM layers negatively affect the representation learning performance (Table 3.7). This is potentially caused by the distorted feature space. The gradients become unstable because the differentiable OT is carried out by iterative matrix inversion. We also observe that the matched pixels are already concentrated (see Fig. 3.3), and thus adding the concentration regularization does not make significant differences.

Table 3.7: Ablation study of differentiable OT, RHM, and concentration loss on PF-PASCAL. OT and RHM are used in training as differentiable layers. The concentration loss is employed as a regularization during training. The results below are evaluated at PCK0.05, and the bold number indicates the best performance.

OT	RHM	Concentration	PCK@0.05
			51.0
✓	✓		32.4
✓			41.5
		✓	48.1

3.4.4 Discussion and Limitation

To better understand how the pixel-level contrastive loss function regularizes the feature representation, we visualize the correspondences given by the affinity matrices on the training dataset. For example, let I_0 and I_1 denote source and target images, and I_0' the patch image randomly cropped from I_0. Fig. 3.4 shows that the cycle constraint can help the cross-instance correspondence prediction of I_0' onto I_1 by enforcing the correspondence predictions of I_0' on I_0 to be the same as the ground truth. For example, the patch image I_0' in the fourth row is cropped from the right part of a juice bottle. Interest-

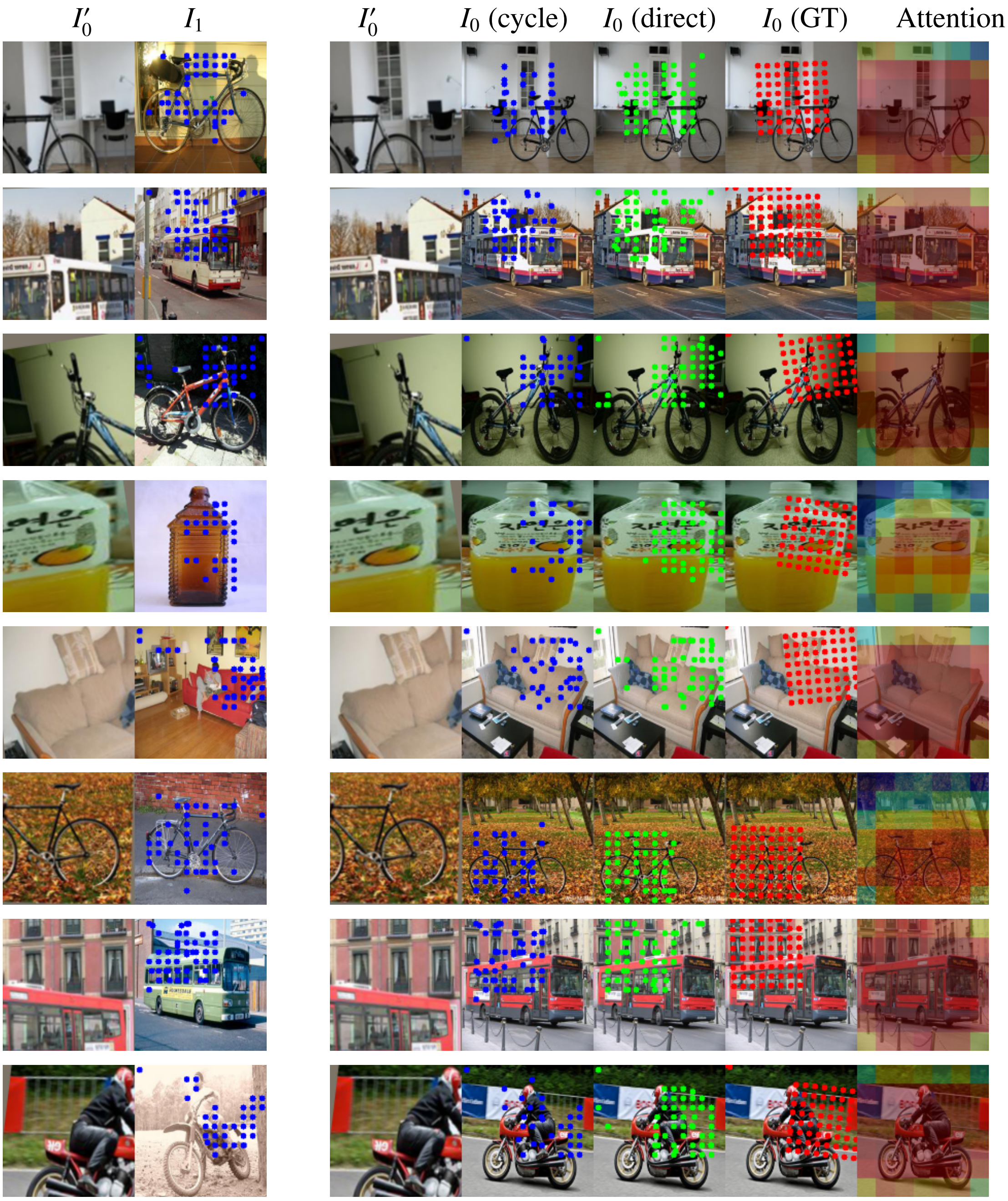


Figure 3.4: Visualization of pixel-level contrastive learning. The first two columns show correspondence predictions across different images (i.e., the blue dots in the second column denote the correspondence of each pixel in I'_0). The remaining five columns show correspondence predictions from the augmented patch images I'_0 to their source images I_0, where the red dots denote the correspondence ground truth (where I'_0 is cropped from I_0), blue or green dots indicate the correspondence prediction based on the cycle affinity matrix $\bar{A}_{0'0} = A_{0'1}A_{10}$ or affinity matrix $A_{0'0}$. The last column shows the self-attention map on which the randomly cropped augmented image I'_0 is based. Note that the colored dots are computed based on the feature map. Thus the number of colored dots is less than the number of pixels.

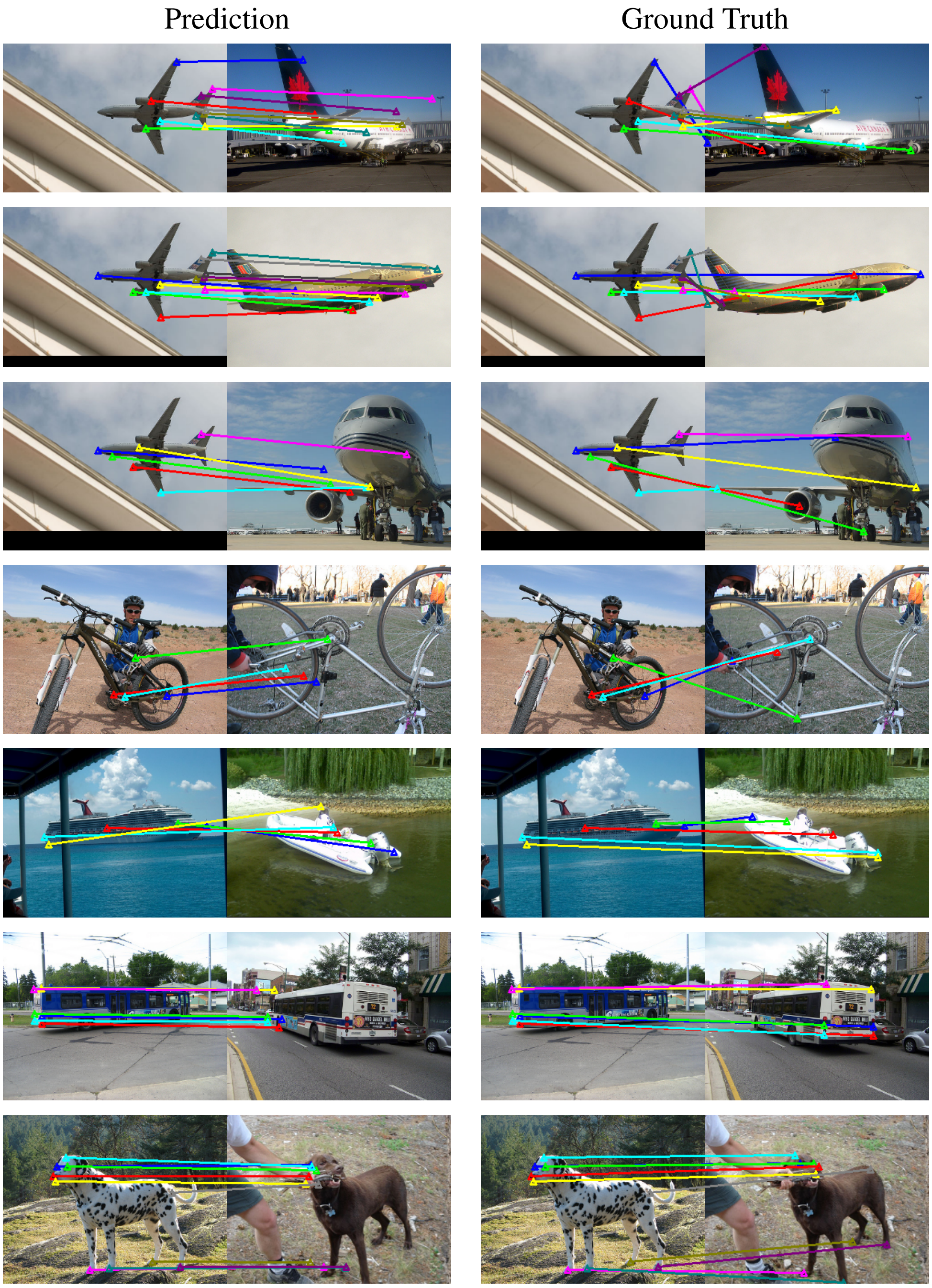


Figure 3.5: Visualization of the correspondence predictions on SPair-71k. The left and right columns are the predictions of our method and the ground truth, respectively. Different keypoints matches are indicated by different colors. The predictions are based on the voting matrix by going through the process of OT and RHM. See Section 3.4.4 for a detailed discussion.

Table 3.8: Evaluation results on the SPair-71k dataset. Numbers in bold indicate the best performance.

Methods	SPair-71k (α_{bbox})		
	PCK@0.05	PCK@0.10	PCK@0.15
HPF [86]	5.50	14.20	22.70
DHPF [87]	5.03	13.71	22.58
SCOT [78]	6.20	15.60	24.60
Ours	**6.30**	**15.80**	**25.20**

ingly, the correspondence prediction on I_1 (the second column) is also on the right side of a different bottle. We observe a similar phenomenon in the other rows. These results demonstrate that multi-level contrastive learning helps focus on the contextual semantic information and learn spatially aware pixel-level feature representations.

Numerous image-level contrastive learning methods have been proposed for object recognition and related tasks in recent years. However, it has not been exploited for learning correspondences especially on the semantic level. We show that image-level contrastive learning facilitates object-centric attention, which in turn helps finding semantic correspondence. Our approach is in direct contrast to existing semantic correspondence methods that mainly use ImageNet pretrained models for feature extraction.

Moreover, the proposed cross-instance pixel-level contrastive learning is specifically designed for semantic correspondence. Existing pixel-level contrastive learning approaches [129, 94, 51] are direct extensions of image-level instance contrastive learning (e.g., MoCo [37]). As such, they compute contrastive losses between different views of the same instance, e.g., either via self-augmented images [129, 94] or from sequences of the same instance [51, 135]. Different from existing contrastive learning approaches, the proposed cross-instance pixel contrastive learning method leverages different images of the same category via cycle consistence. This is not a straightforward extension of the aforementioned approaches, especially when we do not leverage any pixel-level ground truth for constructing positive/negative sample as opposed to the scheme in [58]. The proposed method aims to learn category-level mid-level representations, which is more challenging than the goals of existing approaches.

Table 3.9: Evaluation results on the SPair-71k dataset by using different percentages of pixel correspondence ground truth. For example, 5% GT means that 5% of training image pairs are provided with their pixel correspondence ground truth.

Methods	SPair-71k (α_{bbox})		
	PCK@0.05	PCK@0.10	PCK@0.15
Ours (0% GT)	6.30	15.80	25.20
Ours (5% GT)	6.58	16.03	25.41
Ours (10% GT)	8.51	19.91	30.20
Ours (15% GT)	9.75	22.49	33.56
Ours (100% GT)	11.57	25.83	37.60

We further compare our method with other baselines on the SPair-71k dataset [88], which is a more challenging large-scale dataset. We use the self-supervised pretrained network from MoCo [37] as initialization for all methods. As shown in Table 3.8, our method performs favorably against the HPF [86], DHPF [87] and SCOT [78] approaches.

However, the results from the proposed self-supervised models are slightly worse than those by the supervised methods. To better understand the algorithmic performance, we use the self-supervised pretrained model as initialization and utilize some labeled image pairs with pixel correspondence ground truth for training. As shown in Table 3.9, leveraging pixel correspondence ground truth can improve model performance.

We note that the SPair-71 dataset is significantly more challenging than the PF-PASCAL and PF-WILLOW databases, as there exist large pose variations, viewpoint changes, scale differences, and occlusions in it. Fig. 3.5 presents some of those challenging cases where our method fails to obtain a good result. There are three directions to improve our proposed method. First, a self-supervised pretrained feature extractor needs to be improved to provide more robust feature representations by employing augmentations that can synthesize images with larger scale and viewpoint changes. Second, we may need to combine the cross cycle consistency regularization with a keypoint detector to learn more fine-grained correspondences. Third, image-level contrastive learning could be improved by including more similar instances from different images to construct positive samples. Our future work will focus on addressing these issues.

3.5 Conclusion

We pose a new task to learn semantic correspondence without relying on supervised ImageNet pretrained models or ground truth annotations from the validation set. In the proposed method, we develop a multi-level contrastive learning framework where the image-level contrastive learning module generates object-level discriminative representations, and the pixel-level contrastive learning method constructs fine-grained representations to infer dense semantic correspondence. The pixel-level contrastive learning module is realized through the proposed cross-instance cycle consistency regularization, where we leverage different objects of the same category in different images without knowing their dense correspondence labels. This is different from existing approaches where correspondences are constructed by augmenting the same object in images or video sequences. Experimental results on the PF-PASCAL, PF-WILLOW, and SPair-71k datasets demonstrate the effectiveness of the proposed method over the state-of-the-art schemes for semantic correspondence.

Chapter 4

Exploiting Category Names for Few-Shot Classification with Vision-Language Models

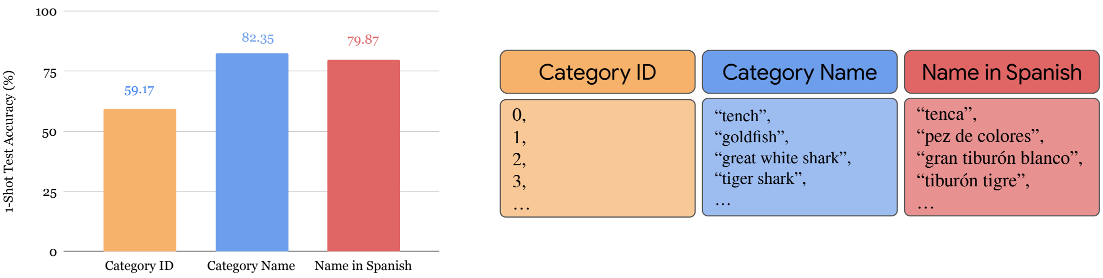
Vision-language foundation models pretrained on large-scale data influence many visual understanding tasks. Notably, many vision-language models build two encoders (visual and textual) that can map two modalities into the same embedding space. As a result, the learned representations achieve good zero-shot performance on tasks like image classification. However, when there are only a few examples per category, the potential of large vision-language models is not fully realized, mainly due to the disparity between the vast number of parameters and the relatively limited amount of training data. This chapter shows that we can significantly improve the performance of few-shot classification by using the category names to initialize the classification head. More interestingly, we can borrow the non-perfect category names, or even names from a foreign language, to improve the few-shot classification performance compared with random initialization. With the proposed category name initialization method, our model obtains state-of-the-art performance on several few-shot image classification benchmarks (e.g., 87.37% on ImageNet and 96.08% on Stanford Cars, both using five-shot learning). Additionally, we conduct an in-depth analysis of category name initialization, explore the point at which the benefits of category names decrease, examine how distillation techniques can enhance the performance of smaller models, and investigate other pivotal factors and intriguing phenomena in the realm of few-shot learning. Our findings offer valuable insights and guidance for future research endeavors.

4.1 Introduction

In recent years, large vision-language models have opened doors to many new applications and provided new thoughts to existing problems. The advantages of large vision-language models are blessed by learning from largely available images with surrounding texts, as well as exploring the capacity of transformer network [21] to model web-scale image-text data. Radford *et al.* [98] first proposed CLIP for vision-language modeling, which was followed by numerous works, including ALIGN [55], LiT [145], Flamingo [2], Florence [142], CoCa [141], etc. The development of vision-language models provides novel perspectives of few-example learning.

This chapter considers the problem of few-shot classification in the new light of large vision-language models. Researchers have found that models pretrained from ImageNet can be easily transferred by finetuning on a new classification task [45]. Similarly, we can take the vision encoder from the pretrained vision-language model and finetune it with a few examples. Since state-of-the-art vision-language models were pretrained on billions of web images and texts, such finetuning often outperforms the models trained on ImageNet with better robustness and generalization capabilities. Moreover, large vision-language models can be adapted to more downstream tasks with fewer labeled data.

Despite the capability of the text branch in pretrained vision-language models, it is not optimally utilized when directly fine-tuning the vision component for downstream image classification tasks. Furthermore, the substantial size of these models may result in overfitting when trained with limited data. In addition to the above approach, we exploit another source of information in vision-language models that traditional models have overlooked. Such new information comes from the category names in downstream image classification tasks. Because vision-language models can generate powerful representations for images and texts, we will show that by utilizing semantic category names for initialization, vision-language models can be transferred better with



Category ID	Category Name	Name in Spanish
0, 1, 2, 3, …	“tench”, “goldfish”, “great white shark”, “tiger shark”, …	“tenca”, “pez de colores”, “gran tiburón blanco”, “tiburón tigre”, …

Figure 4.1: Comparing one-shot classification accuracy on ImageNet using different category information. The typical way of finetuning using images with their category IDs does not work well for one-shot learning with big models. With the information on the category names of training images, we develop a new initialization approach that significantly boosts the performance of vision-language models in few-shot learning. Interestingly, using non-English names can still help even though the model was pre-trained using images and English text data pairs.

few examples in downstream tasks.

As summarized in Figure 4.1, this chapter explores several scenarios: (1) randomly initializing a classification head; (2) initializing a classification head with category names; (3) initializing a classification head with other heuristics such as class digits or even non-English category names. Note that (1) corresponds to the scenario when we only know the category ID (e.g., class 0, class 1, ..., class N) without knowing the meaning of each category. However, (2) implicitly parses the information from category names such as “tench” and “goldfish”. The pretrained language model could process these label names to provide a better initialization for the model adaption. Compared to (2), (3) provides different types of category name information. The main difference between scenario (1) and the others is that (1) does not utilize text/language information from the categories. In scenario (1), the backbone network is initialized from the pretrained model weights, and the classification head is randomly initialized. We set (1) to be our baseline as it is the most common model adaptation method. We leverage the pretrained language model for the other scenarios to parse the text information in the provided categories. Specifically, we pair all category names with prompts and extract the average text embedding as the weight to initialize the classification head. The second scenario is called *category name initialization* (CNI), and it has achieved the best performance among all these scenarios when finetuning using one-shot ImageNet data, as shown in Figure 4.1.

In this chapter, we conduct extensive experiments exploring few-shot performance on ImageNet [18], Cifar100 [66], Oxford Flowers [91], Stanford cars [65], etc. Using the powerful pretrained models, we sweep hyper-parameters such as learning rates, training layers, weight regularization, etc., and find a stable recipe for few-shot learning that can significantly outperform the state-of-the-art in many classification tasks. Notably, we achieve a one-shot top-1 accuracy of 86.15% and a five-shot 87.90% top-1 accuracy on ImageNet, which outperforms many other approaches using the same or more training examples. More interestingly, in this chapter, we demonstrate that:

- Category name initialization can significantly boost the finetuning performance in few-shot settings, outperforming many other initialization or fine-tuning methods. However, the contribution of category names diminishes when there is a sufficiently large number of training images.

- Leveraging the proposed category name initialization can speed up convergence compared to random initialization.

- In scenarios where a user does not speak English, we find that the non-English category name still helps with few-shot learning. For example, we can use Spanish category names to initialize the network, which is more effective than random initialization.

- A larger pretrained model could further boost the few-shot performance of a small model by carrying out model distillation. We have achieved a 1.01% performance boost using 1% labeled images from ImageNet.

- The selection of finetuning layers is crucial to the performance. Empirically, finetuning the last few layers is much better than full model finetuning in a few-shot setting. On the other hand, finetuning the entire network works better when the training data is sufficient.

- We explore additional factors that impact few-shot learning, specifically the learning rate and weight regularization. We provide a comprehensive guide on determining the optimal learning rate and analyze the interesting effects of incorporating L_2 weight regularization into few-shot learning.

4.2 Related Work

The human vision system can surprisingly learn from only a few examples. More amazingly, one may learn more effectively by knowing the new species' names. For example, people who have seen "fish" and "cat" before can quickly understand what "catfish" means with or without the help of additional images. Motivated by this phenomenon, few-shot learning has been extensively studied in computer vision [25, 35]. Since deep CNN became popular, a common practice is to train a deep CNN on ImageNet and then transfer the model to downstream tasks [45]. However, transferring a pretrained ImageNet model requires hundreds or thousands of images. When there are only a few examples per category, the few-shot learning using pretrained ImageNet models is inferior to those trained with enough in-domain data.

Recently, there has been increasing interest in utilizing the vision-language model for visual zero-shot learning, a related problem of few-shot learning. CLIP [98] is a pioneering work in large-scale vision-language modeling. Unlike previous works in vision-language representation [20, 123], CLIP collects image-text pairs from the Web, which contains diversified semantics in a weakly supervised fashion. In addition, CLIP is built on large-scale contrastive learning, which maps images and text into the same subspace. Through this, the model can map textual class names with images hence performing image classification in a zero-shot manner. The approach of CLIP was followed by ALIGN [55], Flamingo [2], LiT [145], Florence [142], FLAVA [110], SimVLM [131] and CoCa [141]. Among these works, ALIGN, Florence, FLAVA, and LIT are based on contrastive learning. Flamingo chooses to optimize a generative loss with gated cross-attention layers. At last, CoCa integrates contrastive and generative loss into one framework. Although training CoCa seems the most challenging among all these vision-language works, it obtains consistently better results in many tasks.

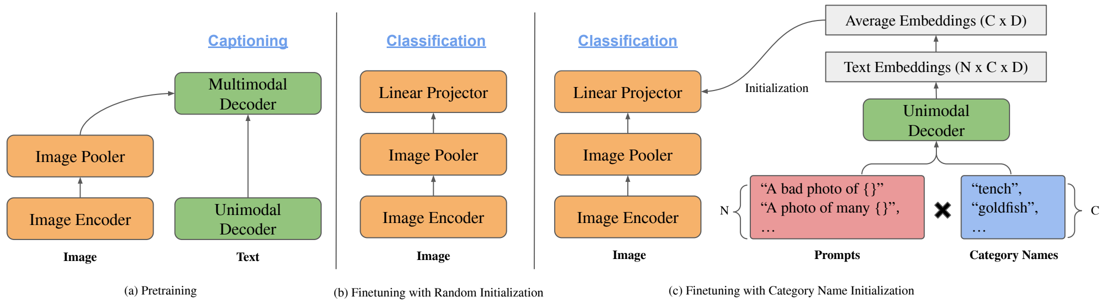
In the literature, CLIP, LiT, ALIGN, Florence, FLAVA, and CoCa have demonstrated promising results with zero-shot learning. However, the potential of these models for few-shot learning is not well exploited. [70] construct a benchmark and toolkit named Elevater for evaluating the transferability of vision-language models using different training samples. [98] point out that using few training examples could improve the effectiveness robustness while undermining the relative robustness. Few-shot learning algorithms are trained exclusively on image data, ignoring the valuable text information that can be used to enhance the learning process. However, Flamingo has emerged as a promising approach for addressing this issue. Flamingo utilizes few-shot interleaved prompts that incorporate gated cross-attention layers to improve few-shot learning.

Zhou *et al.* [149] propose context optimization (CoOp) to model text in prompts through continuous representations. CoCoOp [148] extends CoOp by further learning a lightweight neural network to generate an input-conditional token (vector) for each image. In addition, a series of prior-based methods utilize CLIP priors with a cache model. CLIP-Adapter [26] combines zero-shot visual or language embeddings with corresponding finetuning features to improve performance. TIP-Adapter [146] constructs adapters using a key-value cache model from few-shot training sets and updates their prior knowledge through feature retrieval. TIP-X [118] further constructs an affinity matrix by measuring the KL divergence between test and few-shot samples, which removes direct reliance on the uncalibrated image-image similarities. APE [155] explores the trilateral affinities between the test image, prior cache model, and textual representations and only enables a lightweight category-residual module to be trained. Among these approaches, TIP-Adapter, TIP-X, and APE are training-free, while CoOp, CoCoOp, CLIP-Adapter, and APE-T [155] require training.

Klein *et al.* [64] suggest that using a fisher vector derived from other distributions can improve accuracy in central computer vision tasks. Category names have also been exploited in image-text tasks, such as visual grounding [126] and visual question answering [30]. In these methods, the text embedding of the category names and the image embedding are extracted separately by two branches. Then their inner product is calculated as the similarity score between an image region and an object category. This chapter demonstrates that leveraging category names for initialization can significantly enhance the few-shot performance of the CoCa model without bells and whistles. Our approach outperforms Flamingo and CLIP's performance and establishes a new state-of-the-art for both ImageNet and several other datasets with fewer training examples.

4.3 Approach

In this section, we first briefly review CoCa [141], one of the state-of-the-art vision-language models, and then discuss two initialization strategies: the standard random initialization and new category name initialization (CNI) for finetuning tasks.



(a) Pretraining

(b) Finetuning with Random Initialization

(c) Finetuning with Category Name Initialization

Figure 4.2: An overview of the CoCa pretraining and finetuning. (a) The pretraining of CoCa relies on mapping image and text pairs into the same space for embedding alignment, where the image and text embeddings are extracted through an image encoder and a unimodal text decoder, respectively. The image pooler is used to customize the image embedding for different tasks. (b) We append a randomly initialized linear projector to the image pooler and initialize the image encoder from pretrained weights. (c) We construct text sequences by pairing all C category names with N different prompts. Via the pretrained unimodal decoder, we can compute the text embeddings for all text sequences (with a total number of $N \times C$), each of which is a D-dimensional vector. The normalized average embeddings can be used to initialize the linear projector's weight.

4.3.1 Revisiting CoCa pretraining

Unlike other recent vision-language models, CoCa adopts an encoder-decoder model architecture to learn the generic vision and multi-modal representations. As shown in Figure 4.2 (a), CoCa encodes images to latent representation via an encoder network (e.g., vision transformer (ViT) [21]) and encodes text representations via a unimodal decoder. We append an image pooler after the image encoder to customize the image representations. Practically, CoCa adopts a cascade design by using two image poolers, i.e., a generative image pooler and a contrastive image pooler. The motivation for this design comes from the preliminary experimental results that single pooled image embedding helps vision recognition tasks while more visual tokens benefit multi-modal understanding tasks. Following [68], both generative and contrastive image poolers are single multi-head attention layers with different numbers of learnable queries, enabling the model to pool embedding with different lengths. They can also customize visual representations for different tasks and training objectives. For simplicity and clarity, we depict them in one box named *Image Pooler*. On the other hand, CoCa uses a unimodal decoder to extract text-only embeddings. It cascades multi-modal decoder layers cross-attending to image embeddings to learn multi-modal image-text representations.

CoCa is pretrained on image-text pairs using two objective functions. The first is contrastive loss, where the image representations are contrasted against the paired text representations. The contrastive loss enables cross-modal representation alignment. The other is image-captioning loss, which requires the model to auto-regressively predict the tokenized texts by maximizing the conditional likelihood. The resulting CoCa can thus generate both unimodal visual/textual embeddings and multi-modal joint embeddings. The unimodal visual output generated by the encoder and the unimodal textual output generated by the unimodal decoder are aligned in the same vector space and thus can be used to map images with their class names in a zero-shot manner. Here, we focus on reusing these two components to initialize for few-shot learning.

4.3.2 Finetuning CoCa

Random initialization. One straightforward model adaption approach is to add a randomly initialized linear projector upon the pretrained model and selectively finetune the model (all or part of the layers), as depicted in Figure 4.2 (b). Following the approach used by CLIP [98] and CoCa [141], we first use an image pooler to obtain the aggregated image embedding $H \in \mathcal{R}^D$ and then apply a linear projector to get the prediction

$Y \in \mathcal{R}^C$,

$$Y = \mathrm{softmax}(WH + b), \tag{4.1}$$

where $W \in \mathcal{R}^{C \times D}$ and $b \in \mathcal{R}^C$ are learnable weight and bias of the linear projector. Here W and b are randomly initialized, while the image encoder and generative image pooler are initialized from the pretrained weights. Table 4.7 summarizes the number parameters of different modules of CoCa.

Category name initialization. We argue that the above random initialization ignores the potential of the language model for model adaptation. In contrast, we propose the category name initialization to maximize the capacity of the pretrained unimodal decoder. First, we pair all category names (whose total number is C) with N different prompts as the text inputs. For example, pairing the category name "tench" with a prompt "A bad photo of {}" gives us a text sequence "A bad photo of tench". Next, we compute the text embeddings for all these $N \times C$ text sequences via the unimodal decoder. As the text embedding for each text input is a D-dimensional vector, we can obtain a text embedding tensor with a shape of $N \times C \times D$. Following the previous work CLIP [98], we compute the average over different prompts and perform the L_2 normalization to obtain the average embeddings of shape $C \times D$. Unlike random initialization, we initialize the weight W by the average embeddings and bias b by a zero vector in the linear projector. We initialize the image encoder and the image pooler from the pretrained model weights to enable zero-shot inference of the category name initialized model.

Discussion. Category name initialization is model-agnostic, making it applicable to other foundation models that utilize contrastive loss. Vision-language models trained with contrastive learning inherently yield a two-tower representation, where the text tower's output is embedded into the image tower's embedding space. This shared embedding space allows for cosine distance computation through the inner product of normalized embedding vectors. Consequently, the text embeddings of category names can effectively initialize the visual classifier.

With category name initialization, the model can maintain its zero-shot performance even before fine-tuning, avoiding starting from scratch and undergoing a lengthy fine-tuning process. In contrast to CoOp [149], where prompts are learnable variables, the prompts for each downstream image classification task are fixed. In Section 4.4.3, we will demonstrate that context optimization is less effective than category name initialization. TIP-Adapter [146] calculates predicted logits by measuring the affinity between embeddings of the test image and cached training images, as well as textual embeddings. In Section 4.4.4, we will show that using cached image embeddings for initialization leads to poorer performance. CLIP-Adapter [26] introduces and fine-tunes two learnable adapters, each consisting of two layers of linear transformations, to transform classifier weights and image features. However, we found that using text embeddings of category names to initialize the classifier and finetuning the final few layers is the most effective method for few-shot learning without the need for overly complex designs. Layer selection will be discussed in Section 4.4.7.

In practice, we may not always have the names of all categories. For example, when the finetuning service is provided to users from another country with different languages, the user may use category names in a foreign language or even digital labels for each category. Interestingly, although trained only with English texts, CoCa uses a word piece model and sentence piece model as the tokenizer and thus can compute the embedding for any text sequence without reporting the out-of-vocabulary error. In Section 4.4.4, we will compare the impact of different variants of category name initialization.

4.4 Experiments

In this section, we first describe the details of our experimental setups, and then present our experimental results as well as key findings with comprehensive analysis.

4.4.1 Experimental Setup

Data. We conducted finetuning experiments on several widely-used image classification datasets, including ImageNet [18], ImageNet-V2 [101], ImageNet-R [40], ImageNet-A [41], ImageNet-Sketch [125], Cifar100 [66], Oxford Flowers [91], Stanford Cars [65], Country-211 [98], Food-101 [4], FGVC Aircraft [80], EuroSAT [39], and Oxford-IIIT Pets [96]. To account for different few-shot settings, we randomly sampled a specific portion of data from each dataset. For instance, in one-shot ImageNet, we only chose one image from the ImageNet training data for each category. Despite this sampling, we evaluated all models on the entire testing set. Following the existing benchmark [70], we employed the same text prompts[1] for evaluating all methods for a fair comparison. CoCa [141] is pretrained using JFT-3B [144] and Align datasets [55]. During the pre-training stage, all near-domain examples (3.6M images) are removed following the strict de-duplication procedures [144, 55].

Optimization. We use the Adafactor optimizer [108] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and a weight decay ratio of 0.01. All input images are first rescaled to 580 × 580 and then randomly cropped to the size of 540 × 540. We further apply RandAugment [15] and label smoothing in our data preprocessing pipeline. Our model is implemented in the Lingvo framework using Tensorflow [109].

Hyper-parameters. The choice of batch size depends on the dataset and its number of categories. When the total number of training examples is relatively small, using a large batch size may not be feasible. However, using the largest possible batch size for efficient training is generally desirable. For instance, in the case of ImageNet, which consists of 1000 categories, we opt for a batch size of 512. This decision is based on the consideration that we have a substantial number of images per category, either 1000 images (for one-shot tasks) or 5000 images (for five-shot tasks). Therefore, using a batch size of 512, we can efficiently utilize the available computational resources during training. However, it is important to note that the batch size is adjusted accordingly for datasets with a smaller number of categories. For instance, in the case of Cifar-100, where there are 100 categories, we choose a batch size of 256 for the five-shot setting and 64 for the one-shot setting.

[1] `https://github.com/Computer-Vision-in-the-Wild/Elevater_Toolkit_IC/blob/main/vision_benchmark/datasets/prompts.py`

Model cost. The computational cost of training a model depends on the model size and the chosen training batch size. To provide specific examples, when fine-tuning CoCa-base on ImageNet (five-shot), we utilized a 4x4 Jellyfish TPU with a batch size of 512, and the training process took approximately 6 hours. Similarly, when fine-tuning CoCa-2B on Cifar-100 (five-shot), we employed a 4x4 Dragonfish TPU with a batch size 256, and the training duration was around 9 hours.

4.4.2 Improving CoCa in few-shot classification

State-of-the-art on ImageNet and its variants. We use the pretrained CoCa model and apply category name initialization. We then compare our method against the previous works on ImageNet and its variants, including ImageNet-V2 [101], ImageNet-R [40], ImageNet-A [41] and ImageNet-Sketch [125]. As shown in Table 4.1, CoCa-2B+CNI has achieved state-of-the-art few-shot classification results on all these benchmarks. Surprisingly, the one-shot and five-shot performance of CoCa-base is even better than the performance of some other recent methods finetuned on the whole dataset.

State-of-the-art on other benchmarks. In addition to ImageNet and variants, we show that our method can achieve state-of-the-art few-shot performance on other image classification benchmarks, including Cifar100 [66], Oxford Flowers [91] and Stanford Cars [65], Country-211 [98], Food-101 [4], FGVC Aircraft [80], EuroSAT [39], and Oxford-IIIT Pets [96]. By examining Table 4.2, it becomes apparent that our CoCa-2B model outperforms many other approaches, even when trained with fewer data. The performance gain results from the category name initialization, which serves as a strong foundation that enables the model to achieve better results with only a few examples. To gain a deeper understanding of this phenomenon, we provide an analysis of the category name initialization in the following section.

Table 4.1: Few-shot results on ImageNet and its variants. We use IN as the abbreviation for ImageNet, and CNI for category name initialization. The second column means how much training data per class is used for finetuning. 0 shot means the pretrained vision-language model is directly evaluated without finetuning. Full means the entire training set has been used. All the numbers under the last five columns denote the top-1 test accuracy.

Model	Shot	IN	IN-V2	IN-R	IN-A	IN-Sketch
MAE [36]	full	-	-	66.50	76.70	50.90
CLIP (ViT-B/16) [98]	0	68.40	62.60	77.60	50.00	48.20
	full	79.90	69.80	70.80	46.40	46.90
CLIP (ViT-L/14) [98]	0	76.20	70.10	88.90	77.2	60.20
	full	85.20	75.80	85.30	76.10	58.70
CLIP+Adapter (ResNet-50) [26]	0	55.50	-	-	-	-
	1	58.10	-	-	-	-
	4	59.50	-	-	-	-
CLIP+CoOp (ViT-B/16) [149]	0	58.18	-	-	-	-
	1	58.00	-	-	-	-
	4	60.01	-	-	-	-
Tip-Adapter-F (ResNet-50) [146]	0	60.33	-	-	-	-
	1	61.32	-	-	-	-
	4	62.52	-	-	-	-
WiSE-FT (ViT-L/14) [132]	full	85.30	76.90	89.80	79.70	63.00
Flamingo-3B [2]	1	70.90	-	-	-	-
	5	72.70	-	-	-	-
Flamingo-80B [2]	1	71.90	-	-	-	-
	5	77.30	-	-	-	-
CoCa-base [141]	0	82.26	76.22	93.16	76.17	71.12
CoCa-base+CNI (Ours)	1	82.35	76.47	93.37	77.00	71.61
	5	83.58	77.23	93.22	77.23	71.35
CoCa-2B [141]	0	86.09	80.39	96.19	89.39	77.12
CoCa-2B+CNI (Ours)	1	86.15	80.57	96.62	90.12	77.49
	5	87.37	81.66	96.41	89.68	77.39

Table 4.2: Comparing with the state-of-the-art on multiple classification benchmarks. CNI stands for category name initialization, and RI means random initialization. Our model obtains the state-of-the-art few-shot learning performance with less training data than others.

Model	Shot	Cifar100	Oxford	Stanford	Country-211	Food-101	FGVC	EuroSAT	Oxford-IIIT
			Flowers	Cars			Aircraft		Pets
MAE [36]	5	21.20	50.90	6.30	2.80	7.70	7.00	64.60	17.20
	20	43.50	71.90	25.50	4.40	30.40	29.90	74.10	60.00
	full	68.30	72.00	37.20	10.10	65.10	39.10	94.80	81.60
CAE [10]	5	38.30	70.30	8.70	3.50	18.60	14.30	76.70	37.30
	20	55.10	81.20	27.50	5.50	35.70	32.60	89.00	63.30
	full	78.90	81.20	40.40	11.40	67.40	40.80	96.70	79.80
MoCo-v3 [11]	5	60.50	79.50	13.40	4.80	36.60	11.80	77.10	76.20
	20	75.50	89.50	49.50	7.60	59.30	38.20	84.80	86.40
	full	85.30	89.50	63.00	13.70	78.00	48.00	95.90	91.40
DeiT [116]	5	61.50	82.70	27.60	4.40	41.90	24.10	62.50	87.80
	20	73.70	92.70	68.80	6.20	61.50	34.10	90.70	91.90
	full	89.60	92.40	83.00	14.10	84.50	59.30	98.20	93.90
ViT [21]	5	75.40	99.20	27.60	6.80	59.00	22.70	70.00	89.60
	20	84.00	99.20	53.90	11.50	81.70	40.50	86.50	92.60
	full	89.80	99.20	67.50	16.60	89.60	47.80	96.00	94.80
CLIP [98]	5	71.10	94.20	73.60	21.70	89.70	36.00	76.70	90.50
	20	75.40	96.8	73.60	25.20	90.60	48.10	86.60	92.30
CoCa-2B [141]	0	77.19	92.04	94.37	42.15	94.79	44.83	49.74	97.88
CoCa-2B+RI	1	5.69	40.78	14.29	1.71	1.26	12.24	56.84	61.95
	5	7.49	84.71	86.31	19.06	62.45	27.21	82.38	78.61
CoCa-2B+CNI	1	77.89	98.45	95.29	42.44	94.91	58.33	75.06	97.93
	5	78.62	99.25	96.08	44.52	95.50	69.29	85.78	98.12

4.4.3 Analysis of category name initialization

This section delves deeper into how the proposed category name initialization helps with large vision-language models in few-shot learning. Vision-language models are adept at zero-shot inference without knowing any class names from downstream tasks. However, the zero-shot performance heavily depends on the domain gap and data distribution, thus varying on different downstream tasks. By leveraging a few training examples from the target domain, the pretrained vision-language models can adapt to the target domain.

Improvement upon zero-shot performance. We first examine how category name initialization improves zero-shot performance. As illustrated in Table 4.1 and Table 4.2, category name initialization enhances performance across all datasets. The improvement in performance from zero-shot to five-shot varies depending on the dataset. For instance, CoCa-2B on ImageNet sees a 1.32% increase in performance, whereas EuroSAT sees 36.04% growth. CoCa's zero-shot performance on ImageNet leaves less room for few-shot learning. Nonetheless, the performance gain achieved through our category name initialization is noteworthy, as some other methods may not achieve comparable improvements, which will be discussed below. We also contend that our few-shot performance is not solely attributable to the strong pretrained CoCa model but also our proposed category name initialization. For example, CoCa-2B's zero-shot performance on EuroSAT is 49.74%, which is lower than that of most other approaches. However, with our category name initialization, it achieves 85.78%, outperforming other approaches in the five-shot setting.

Comparing with other fine-tuning methods. To further validate the efficacy of category name initialization, we compare it with several other finetuning methods. We choose CoCa-base as the pretrained vision-language model and carry out experiments on ImageNet with different finetuning methods, such as linear probing, full finetuning, CoOp [149], and category name initialization. As demonstrated in Table 4.3, all finetuning methods, except category name initialization, fail to improve over zero-shot CoCa when one or five training examples per class are used. Furthermore, full finetuning underperforms linear probing because the number of training examples is inconsistent with the number of trainable parameters in few-shot learning. Although showing better performance than linear probing and full finetuning, the one- or five-shot performance of CoOp is slightly inferior to zero-shot CoCa. This suggests that learning contextual prompts does not significantly improve CoCa's few-shot performance. On the other hand, category name initialization effectively improves the few-shot performance, which is challenging when the zero-shot performance of CoCa is significantly higher than that

Table 4.3: Comparing other fine-tuning methods on ImageNet and its variants. We use IN as the abbreviation for ImageNet, and CNI for category name initialization. The second column means how much training data per class is used for finetuning. 0 shot means the pretrained vision-language model is directly evaluated without finetuning. All the numbers under the last five columns denote the top-1 test accuracy.

Model	Shot	IN	IN-V2	IN-R	IN-A	IN-Sketch
CoCa-base	0	82.26	76.32	93.16	76.17	71.43
CoCa-base+Linear Probing	1	57.49	54.20	69.19	53.38	47.94
	5	79.33	73.18	90.02	73.18	68.03
CoCa-base+Full Fintuning	1	43.77	41.64	55.98	40.31	33.29
	5	60.90	54.32	71.20	54.34	49.25
Coca-base+CoOp	1	79.85	73.21	89.88	76.42	65.81
	5	81.01	75.81	92.58	76.55	71.27
CoCa-base+CNI	1	82.35	76.47	93.37	77.00	71.61
	5	83.47	77.23	93.22	77.23	71.35

of other counterparts such as CLIP [98] and FLAVA [110].

Category name initialization vs. random initialization. To gain a deeper understanding of the advantages of category name initialization, we compared it with random initialization. Comparing the last three rows in Table 4.2, we can observe that the few-shot classification results using random initialization are worse than the zero-shot classification with pretrained CoCa. However, employing category name initialization would effectively use those few training examples and boost performance. Figure 4.3 provides a more detailed comparison of the optimization process using the two initialization methods. By meticulously tuning the parameters, we set the initial learning rate to 1e-5 for category name initialization and 5e-5 for random initialization. Employing category name initialization results in a better starting model with higher test accuracy than random initialization. Furthermore, the model utilizing category name initialization converges faster than random initialization. This can be attributed to the fact that the test

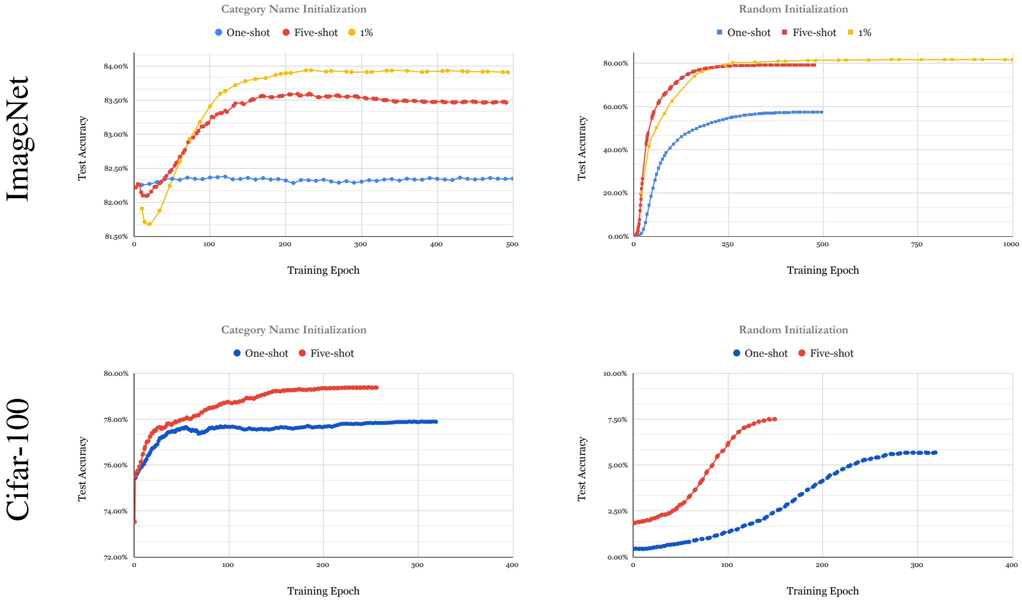


Figure 4.3: Comparison of test accuracy over the training epoch. We finetune the CoCa-base model with category name initialization or random initialization. Category name initialization provides better initial test accuracy and helps the model converge better and faster than random initialization.

accuracy while using random initialization continues to increase even after 250 epochs, whereas the accuracy achieved with category name initialization plateaus around 200 epochs when fine-tuning on ImageNet. Similarly, the one-shot test accuracy on Cifar-100 converges within 100 epochs by employing category name initialization, while the counterpart using random initialization converges after 300 epochs.

4.4.4 Exploring different initialization approaches

In real-world scenarios, we cannot always guarantee the availability of perfect category names for every classification task. Sometimes we may only have digital labels such as class “1”, “2”, and so on, while in other cases, users may not be fluent in English. In such scenarios, it is crucial to evaluate how the model performs with different versions of category names.

Table 4.4 compares the performance of using no category names (i.e., random initial-

Table 4.4: Comparison of category name initialization using digits or different languages. We use the same pretrained CoCa-base model for all category name initialization. The numbers below are top-1 test accuracy on ImageNet.

Category Name Initialization	Zero-shot	One-shot	Five-shot
No	N/A	59.17	79.33
Digits	0.10	53.60	78.75
Korean	22.89	53.71	79.53
Russian	43.59	53.43	79.55
Germany	29.24	63.15	79.90
Spanish	34.38	79.87	80.05
English	82.26	82.35	83.58

ization) with various variants of category names. The most straightforward approach is to use digits (class 1, 2, and so on) as category names. However, this approach provides little semantic information and does not improve few-shot performance. Conversely, category names in English and other languages significantly enhance few-shot recognition. This is surprising because CoCa was trained on English-only text with limited knowledge of other languages. Nevertheless, due to the sentence piece tokenizer [67] and token sharing, our method can still benefit from foreign language transfer, resulting in better performance than random initialization, even though the performance of these foreign language names is not as good as that of English names.

Inspired by the aforementioned observation, we hypothesize that initialization with only partial category information can still yield benefits. To test this hypothesis, we randomly selected 50% of the category names for initialization while using random initialization for the remaining names. The results are shown in Table 4.5, where it can be seen that using 50% of the names still improves the one-shot accuracy from random initialization from 59.17% to 66.82%, and the five-shot accuracy from 79.33% to 80.67%. This indicates that our method has the potential as a valuable tool in situations where within-domain labels are incomplete or expressed in different languages.

Table 4.5: Comparing the performance of using all category names or 50% of names (the other half will be initialized with random vectors) for initialization. The numbers below are top-1 test accuracy on ImageNet.

Initialization	Zero-shot	One-shot	Five-shot
No category name	N/A	59.17	79.33
50% category names	44.36	66.82	80.67
100% category names	82.26	82.35	83.58

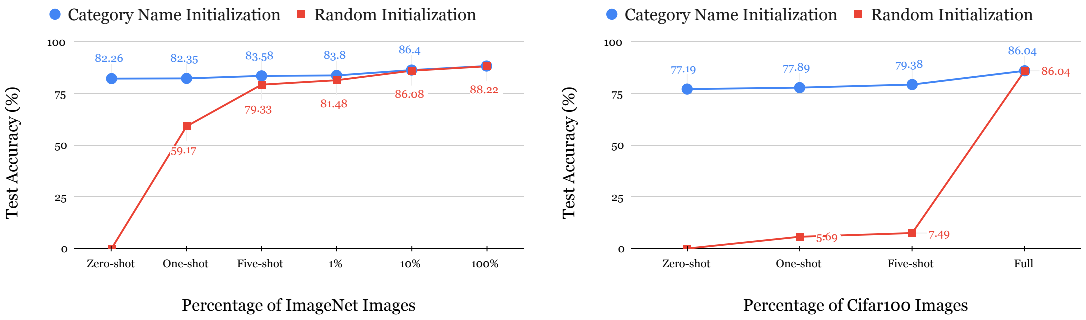
Another question that arises is whether we can apply a similar initialization approach using image embeddings instead of text. To test this hypothesis, we select one representative image per class from ImageNet, resulting in 1000 images for 1000 categories, and used the pretrained CoCa-base model to extract 1000 embedding vectors. We then initialize the linear projector of our few-shot model with these image embeddings, which we call image embedding initialization (IEI). We compare the performance of IEI (using one example image per category) with CNI (using category names but no images) and present the accuracy of initialized models (without finetuning) in Table 4.6. The results indicate that IEI performs worse than CNI, suggesting that embedding category names are more robust than embedding a single image. Moreover, we compute the average of the IEI and CNI weights to create a new initialization vector and find that the average weight's performance lies in the middle of IEI and CNI.

Table 4.6: Comparing top-1 accuracy of image embedding initialization (IEI) and category name initialization (CNI) on ImageNet.

Initialization	Accuracy (%)
IEI	47.16
0.5 × IEI + 0.5 × CNI	61.84
CNI	82.26

4.4.5 Limitations

After comparing different initialization approaches, one question that arises is whether category name initialization continues to be helpful with more training data. We investigate this by fine-tuning pretrained vision-language models using varying numbers of training images. To demonstrate the effectiveness, we establish a baseline for comparison by using random initialization. We utilize two different pretrained CoCa models, CoCa-base and CoCa-2B, and fine-tune them on ImageNet and Cifar100 using different training data. As shown in Figure 4.4, category name initialization outperforms random initialization across different datasets, model architectures, and numbers of training data. However, the contribution of category name initialization diminishes as more training data is provided.



(a) CoCa-base + CNI on ImageNet

(b) CoCa-2B + CNI on Cifar100

Figure 4.4: Comparison of test accuracy over different percentages of training images. Category name initialization outperforms random initialization over different datasets, model architectures, and numbers of training data.

Another limitation of the proposed category name initialization is that it relies on category names to initialize the classification head. While it can significantly improve few-shot image classification accuracy, it may not be applicable in all scenarios. For example, in domains where category names are not available or are not reliable, the proposed method may not be effective.

4.4.6 Model distillation

We first show that category name initialization can be used for different scales of models by carrying out few-shot experiments using two different pretrained CoCa architectures: CoCa-base and CoCa-2B, under different numbers of training data. Abandoning the uni-modal and multi-modal text decoders, CoCa-base and CoCa-2B contain 96M and 1B parameters for downstream image classification tasks (see Table 4.7). As shown in Table 4.8, we can observe the trend that bigger models do better and more shots help.

Table 4.7: Number of parameters of different modules.

Module	CoCa-base	CoCa-2B
Image encoder	85,999,872	1,011,740,288
Image pooler	19,095,296	63,843,648
Linear projector	769,000	1,409,000

Table 4.8: Few-shot results of different CoCa-models on ImageNet.

Model	Zero-shot	One-shot	Five-shot	1%
CoCa-2B	86.19	86.15	87.37	87.90
CoCa-base	82.26	82.35	83.58	83.80
+ distillation	-	-	-	84.81

As larger models tend to perform better, it is natural to consider knowledge distillation, which involves using the predictions of a teacher model to guide the training of a student model. In this work, we use the finetuned CoCa-2B model with 1% of the ImageNet images as the teacher model and CoCa-base as our student model. In addition to the 1% labeled ImageNet images, we use other unlabeled images for knowledge distillation. During the finetuning process, we freeze the teacher model weights and update the student model weights using two loss objectives. The first objective is the supervised loss, where we compute the cross entropy between the student model predictions and the labels for the 1% labeled ImageNet images. The second objective is the distillation loss, computed over all unlabeled data. Unlike few-shot finetuning, where only the last few layers are finetuned, we finetune the entire student model here since the distillation loss is computed over many unlabeled images. For example, table 4.8 shows that by distilling from the larger finetuned teacher model, CoCa-base achieves a 1.01% improvement in accuracy (from 83.80% to 84.81%).

4.4.7 Ablation studies

In this section, we analyze several important factors that influence the few-shot performance. We conduct our ablation study using CoCa-base as the model.

Finetuning layers. We evaluate the performance of the CoCa-base model on ImageNet [18] in various few-shot learning scenarios, with different finetuning layers selected. We compare the results to a baseline using random initialization. In our notation, P denotes the image pooler and L denotes the linear projector. For both category name initialization and random initialization, we experiment with three different optimization strategies: 1) optimizing only the linear projector (L); 2) optimizing both the image pooler (P) and the linear projector (L); and 3) optimizing all layers. Note that we have extensively tried various hyper-parameters (such as initial learning rate) and presented the optimal values for each setting.

The results presented in Table 4.9 indicate that the best performance is achieved by finetuning both the image pooler and linear projector under all settings when compared to the other two optimization strategies for random initialization.

To enhance the few-shot learning performance, we experiment with category name initialization discussed in Section 4.3.2. In contrast to random initialization, we initialize the linear projector using the average text embeddings of the category names. As shown in Table 4.10, this initialization method significantly improves few-shot recognition performance. Moreover, we observe that finetuning P + L is the most effective optimization strategy for few-shot settings while finetuning all layers performs better with more training data.

Table 4.9: Comparison of different finetuning layers for random initialization. P: image pooler; L: linear projector; All: all layers. The best performance of each column is in **bold**.

Finetuning Layers	One-shot	Five-shot	1%	100%
L	49.38	69.64	76.53	85.62
P + L	**57.49**	**79.33**	**81.48**	**88.22**
All	43.77	60.90	79.75	86.03

Table 4.10: Comparison of different finetuning layers for category name initialization. P: image pooler; L: linear projector; All: all layers. The best performance of each column is in **bold**.

Finetuning Layers	One-shot	Five-shot	1%	100%
L	82.35	81.03	81.67	86.16
P + L	**82.35**	**83.58**	**83.91**	88.25
All	82.28	82.63	83.63	**88.35**

Learning rates. We analyze the influence of the initial learning rate on few-shot learning. We set a batch size of 512, froze the image encoder, and adopted a cosine learning rate schedule for the final three layers. Figure 4.5 presents the top-1 test accuracy on ImageNet using different initial learning rates. A small initial learning rate (5e-6) results in a slow convergence rate, while a larger learning rate (5e-5) achieves faster convergence. However, despite reaching the highest test accuracy within 1000 training steps, the finetuning becomes unstable as the test accuracy declines right after the peak value. Conversely, using an even larger learning rate (5e-4) could prevent the surging phase, resulting in a downward trend of test accuracy. By contrast, selecting an appropriate learning rate (1e-5) is the key to stable and rapid few-shot finetuning. Unfortunately, there is no mathematical formula for determining the optimal initial learning rate since it varies across different datasets and depends on the batch size. We can adjust the initial

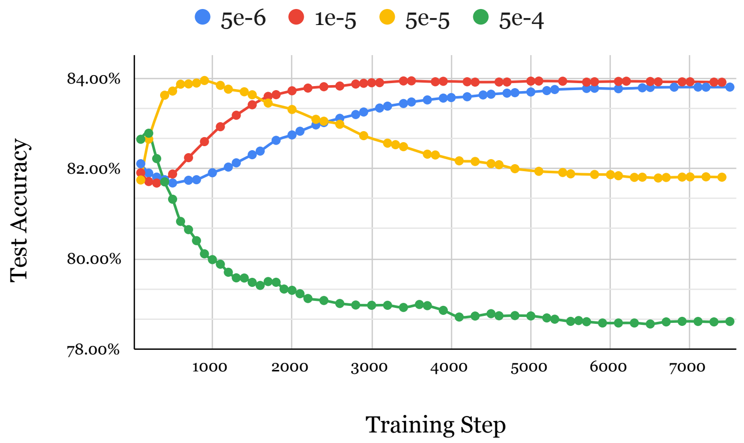


Figure 4.5: The top-1 test accuracy of finetuning CoCa-base on 1% ImageNet using different initial learning rates.

learning rate by trial and observation, and these four test accuracy curves could indicate whether to enlarge or reduce the initial learning rate.

L_2 weight regularization. Out of all the few-shot settings, one-shot learning is the most unique and intriguing. As illustrated in Figure 4.6, the one-shot test accuracy (in red) on ImageNet decreases even with category name initialization during finetuning, unlike the five-shot accuracy (in blue). Using only one training image per class can easily distort the decision boundary, as illustrated in Figure 4.7. We plot the decision boundary in Figure 4.7 for an illustration. Without L_2 regularization, the decision boundary of the finetuned model is easily distorted by the limited training examples, resulting in a degradation from zero-shot performance. However, by applying L_2 weight regularization for one-shot learning, the decision boundary does not deviate much from the decision boundary of the pretrained model. This is reflected in the steady increase of test accuracy from 82.26% to 82.35%, as depicted by the yellow curve in Figure 4.6. Although the performance gain is small, it is still noteworthy since the information provided by one-shot data is limited in helping a pretrained model. On the other hand, applying L_2 weight regularization in five-shot learning could adversely affect the model adaptation, as shown by the green curve. The reason is that L_2 weight regularization, acting as an additional constraint, restricts the model from learning new knowledge from

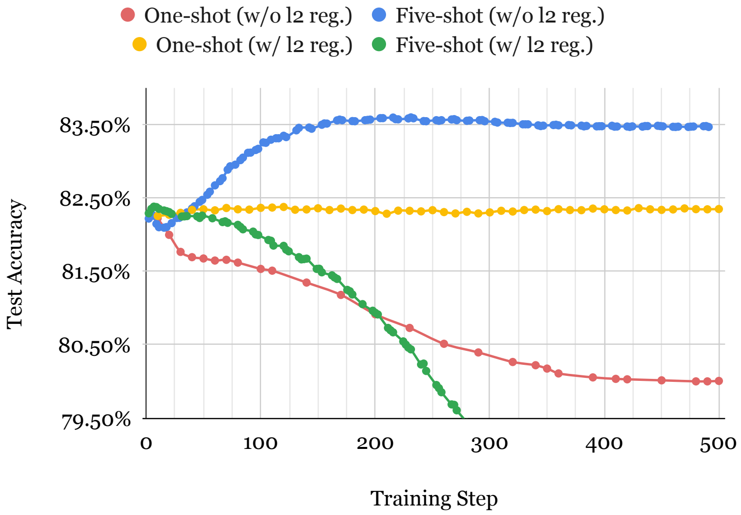
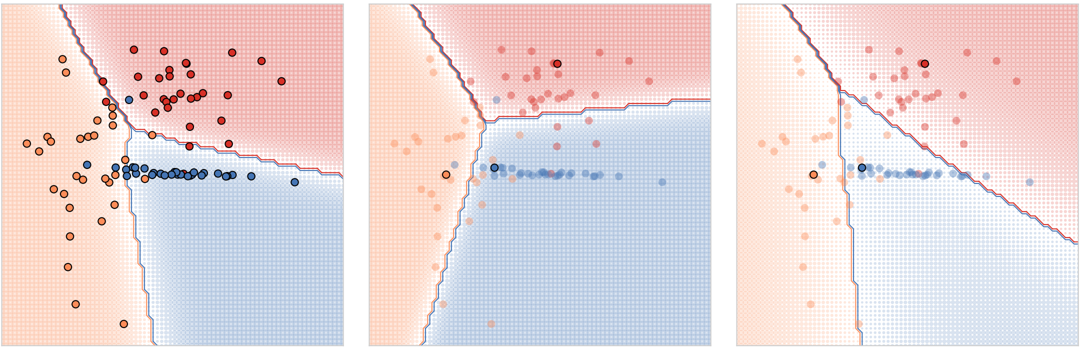


Figure 4.6: The effect of L_2 weight regularization for one-shot and five-shot learning. We plot the top-1 test accuracy of CoCa-base on ImageNet vs. the training step. The L_2 weight regularization is beneficial to one-shot learning but harmful to five-shot learning.

the training data when sufficient information is available to refine the decision boundary of the pretrained model. It should be noted that all of the aforementioned phenomena are dependent on utilizing category name initialization. The decision boundary will lack discriminative power if category name initialization is not used. Therefore, adding L_2 weight regularization would have no meaningful effect.

4.5 Conclusion

This chapter has studied the few-shot classification problem using large vision-language models. Since it is hard to optimize large vision-language models with a few training examples, we propose exploring category names to initialize the classification head, significantly improving performance. In addition, we have also investigated the condition when the category names help. We demonstrate that borrowing other non-perfect category names or even names from a foreign language could also help the few-shot classification of vision-language models, which is better than randomly initializing the classification head. However, the contribution of category names diminishes when



(a) Pretrained model (b) No L_2 regularization (c) L_2 regularization

Figure 4.7: Visualization of decision boundary in one-shot learning. From left to right, The first subfigure displays the decision boundary of the pretrained model. In contrast, the second and third subfigures show the finetuned model without and with L_2 weight regularization, respectively. Each model was trained using only one training example per class, with three classes retained for simplicity. The decision boundary does not shift significantly when finetuning on the one-shot dataset with L_2 regularization. This indicates that the model's generalization ability is improved, as it is less likely to overfit the training examples.

the number of training samples becomes large. This chapter obtains state-of-the-art few-shot performance on numerous benchmarks, including ImageNet, ImageNet-V2, ImageNet-R, ImageNet-A, ImageNet-Sketch, Cifar100, Oxford Flowers, Stanford Cars, Country-211, Food-101, FGVC Aircraft, EuroSAT, and Oxford-IIIT Pets. Our few-shot classification result is even better than many previous works that have employed the whole training set.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

In this thesis, we propose different learning methods for learning three different types of correspondence. Specifically, we have made several key contributions:

Spatial-temporal correspondence in videos. In Chapter 2, we propose a learnable cost volume module for accurate and robust optical flow estimation. This module generalizes the standard inner product by a positive definite kernel matrix. We perform spectral decomposition on the kernel matrix and re-parameterize it via the Cayley representation for efficient optimization. Our module is designed to seamlessly integrate into existing models, effortlessly replacing the standard cost volume. This integration elevates the accuracy of optical flow estimation, bolstering the model's overall robustness and opening up new possibilities for enhanced optical flow applications.

Semantic Correspondence in Images. In Chapter 3, we have established a multi-level contrastive learning framework to learn semantic correspondence without relying on supervised ImageNet pretrained models or ground truth annotations from the validation set. In our framework, image-level contrastive learning generates object-level discriminative representations, and pixel-level contrastive learning further facilitates fine-grained representations at region- or pixel-level to improve dense semantic correspondence performance. We propose a cross-instance cycle consistency regularization to learn a discriminative local feature at the pixel level without dense ground truth.

Multimodal Correspondence between Images and Texts. In Chapter 4, we have proposed category name initialization for addressing the few-shot image classification problem using pretrained vision-language models. Specifically, we compute the text embeddings of category names and initialize the classification head. This method substantially improves the few-shot performance and accelerates the convergence process across diverse datasets. Furthermore, we investigate various influential factors in the context of few-shot learning, such as selecting fine-tuning layers, learning rate choices, and applying weight regularization. This comprehensive analysis sheds light on the nuances of the few-shot learning process, offering valuable insights into the efficacy of our approach and its adaptability to different settings.

5.2 Future Work

We have explored three different types of correspondence in this thesis. Along the way, we have a myriad of interesting future directions.

5.2.1 Recurrent Distillation for Spatial-Temporal Correspondence

The recent RAFT [115] method has dramatically improved the performance by a large margin compared with existing state-of-the-art techniques. This notable improvement has ignited interest in iteratively refining flow estimation. RAFT operates by extracting per-pixel features, constructing multi-scale 4D correlation volumes for pixel pairs, and updating the flow field iteratively through a recurrent unit that performs lookups on these correlation volumes. This iterative refinement strategy deviates from the conventional pyramid design, such as the widely-used one by Sun et al. [112], resulting in a more compact and efficient model. However, it's important to note that the efficacy of RAFT in unsupervised settings, where flow estimation is substantially more challenging than in supervised scenarios, remains unverified.

In contrast, DDFlow [76] and SelFlow [77] have shown promising results in unsupervised scenarios. Both approaches leverage model distillation, utilizing the flow estimated by a teacher model to guide a student model. The teacher model is trained using raw video frames, and synthetic occlusions are introduced when training the student model. The critical insight here is that the student model can benefit from the teacher model's estimated flow in guiding it through newly generated occluded regions.

Could we employ a similar recurrent distillation approach for flow estimation, akin to the RAFT model's iterative refinement? Without ground truth data, recurrent refinement could be achieved through a teacher-student model distillation process. In each refinement step, the student model is guided by the predictions of the teacher model, gradually improving the accuracy of the estimated flow over multiple iterations.

5.2.2 Leveraging NeRF for Occlusion-Aware Optical Flow Estimation

To address the challenge of occlusion in optical flow estimation, one promising avenue is the utilization of NeRF (Neural Radiance Fields) [85]. NeRF is a cutting-edge technique that leverages neural networks to model the 3D geometry and radiance of a scene from a collection of 2D images. By employing NeRF, we can overcome occlusion issues by reconstructing the 3D structure of the scene and discerning the occluded regions more accurately. This 3D scene understanding enables us to refine optical flow estimation, particularly in complex scenarios where objects interact, occlude one another, or move within the scene. The integration of NeRF with correspondence models offers the potential to greatly improve optical flow accuracy and reliability, making it an exciting avenue for further exploration in the realm of correspondence learning.

5.2.3 Fine-Grained Correspondence and Specialized Tasks

One exciting direction is to explore the fine-grained level of correspondence. Fine-grained correspondence entails precisely aligning and recognizing subtle differences within objects or scenes, often involving similar categories with subtle distinctions. Such fine-grained correspondence can also refer to the images and texts, which requires vision-language models to better understand images based on the texts.

Fine-grained correspondence is pivotal in many applications, such as fine-grained object recognition, biomedical image analysis, and visual grounding. Fine-grained object recognition discriminates objects within the same general category but with subtle differences. In the biomedical field, fine-grained correspondence is indispensable for analyzing medical images. Radiologists and researchers rely on precise correspondence techniques to identify subtle anomalies or differences in medical scans. Based on a natural language query, visual grounding aims to locate the most relevant object or region in an image. A model needs to establish context-level semantic correspondences across the two modalities since the target object is distinguished from other objects based on its visual context (i.e., attributes and relationship with other objects) and correspondences with the semantic concepts of the textual description.

5.2.4 Correspondence in More Modalities

Expanding correspondence to more modalities is a significant and evolving area. One considerable extension of correspondence learning involves the integration of audio modalities. This integration is particularly relevant in video analysis and content understanding applications. Models capable of learning correspondences between visual and auditory information can enable tasks such as sound source localization in videos, audio-visual event detection, and automatic video captioning with audio descriptions. Incorporating haptic and tactile modalities into correspondence learning extends the understanding of the sense of touch in computer vision. Correspondence models that recognize the physical interaction between objects and surfaces can be applied in robotics for object manipulation, grasping, and material recognition. This interdisciplinary approach enhances robots' interaction and understanding of the physical world. In the automotive industry, integrating vision, language, and sensor modalities is key for developing safe and efficient autonomous vehicles. These vehicles use visual data from cameras, textual information for navigation, and sensor modalities like LiDAR and radar to navigate and make real-time decisions. Advanced correspondence models are crucial in understanding the complex driving environment, identifying obstacles, and ensuring passenger safety.

Bibliography

[1] P-A Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.

[2] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning. In *Neural Information Processing Systems (NeurIPS)*, 2022.

[3] Wenbo Bao, Wei-Sheng Lai, Chao Ma, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. Depth-aware video frame interpolation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[4] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 - mining discriminative components with random forests. In *European Conference on Computer Vision (ECCV)*, 2014.

[5] Hilton Bristow, Jack Valmadre, and Simon Lucey. Dense semantic correspondence where every pixel is a classifier. In *IEEE International Conference on Computer Vision (ICCV)*, pages 4024–4031, 2015.

[6] Thomas Brox and Jitendra Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE Transactions on Pattern Recognition and Machine Intelligence (PAMI)*, 33(3):500–513, 2010.

[7] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *European Conference on Computer Vision (ECCV)*, 2012.

[8] Arthur Cayley. About the algebraic structure of the orthogonal group and the other classical groups in a field of characteristic zero or a prime characteristic. *Reine Angewandte Mathematik*, 32:1846, 1846.

[9] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning (ICML)*, 2020.

[10] Xiaokang Chen, Mingyu Ding, Xiaodi Wang, Ying Xin, Shentong Mo, Yunhao Wang, Shumin Han, Ping Luo, Gang Zeng, and Jingdong Wang. Context autoencoder for self-supervised representation learning. *ArXiv*, 2202.03026, 2022.

[11] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *IEEE International Conference on Computer Vision (ICCV)*, pages 9620–9629, 2021.

[12] Yun-Chun Chen, Po-Hsiang Huang, Li-Yu Yu, Jia-Bin Huang, Ming-Hsuan Yang, and Yen-Yu Lin. Deep semantic matching with foreground detection and cycle-consistency. In *Asian Conference on Computer Vision (ACCV)*, pages 347–362. Springer, 2018.

[13] Jingchun Cheng, Yi-Hsuan Tsai, Shengjin Wang, and Ming-Hsuan Yang. Segflow: Joint learning for video object segmentation and optical flow. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[14] Christopher B Choy, JunYoung Gwak, Silvio Savarese, and Manmohan Chandraker. Universal correspondence network. In *Neural Information Processing Systems (NeurIPS)*, 2016.

[15] Ekin Dogus Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. Randaugment: Practical automated data augmentation with a reduced search space. *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 3008–3017, 2020.

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

[17] Kevin Dale, Micah K Johnson, Kalyan Sunkavalli, Wojciech Matusik, and Hanspeter Pfister. Image restoration using online photo collections. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2217–2224, 2009.

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

[19] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1422–1430, 2015.

[20] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2625–2634, 2015.

[21] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

[22] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[23] Olivier Duchenne, Armand Joulin, and Jean Ponce. A graph-matching kernel for object categorization. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1792–1799, 2011.

[24] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal on Computer Vision (IJCV)*, 111:98–136, 2014.

[25] Li Fei-Fei, Robert Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE Transactions on Pattern Recognition and Machine Intelligence (PAMI)*, 28(4):594–611, 2006.

[26] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *International Journal on Computer Vision (IJCV)*, 2023.

[27] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[28] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations (ICLR)*, 2018.

[29] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. In *Neural Information Processing Systems (NeurIPS)*, 2020.

[30] Tanmay Gupta, Kevin J. Shih, Saurabh Singh, and Derek Hoiem. Aligned image-word representations improve inductive transfer across vision-language tasks. In *IEEE International Conference on Computer Vision (ICCV)*, pages 4223–4232, 2017.

[31] David Hafner, Oliver Demetz, and Joachim Weickert. Why is the census transform good for robust optic flow computation? In *International Conference on Scale Space and Variational Methods in Computer Vision (SSVM)*. Springer, 2013.

[32] Bumsub Ham, Minsu Cho, Cordelia Schmid, and Jean Ponce. Proposal flow. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3475–3484, 2016.

[33] Bumsub Ham, Minsu Cho, Cordelia Schmid, and Jean Ponce. Proposal flow: Semantic correspondences from object proposals. *IEEE Transactions on Pattern Recognition and Machine Intelligence (PAMI)*, 40:1711–1725, 2018.

[34] Kai Han, Rafael S Rezende, Bumsub Ham, Kwan-Yee K Wong, Minsu Cho, Cordelia Schmid, and Jean Ponce. Scnet: Learning semantic correspondence. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[35] Bharath Hariharan and Ross Girshick. Low-shot visual recognition by shrinking and hallucinating features. In *IEEE International Conference on Computer Vision (ICCV)*, pages 3018–3027, 2017.

[36] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Doll'ar, and Ross B. Girshick. Masked autoencoders are scalable vision learners. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15979–15988, 2022.

[37] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9726–9735, 2020.

[38] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[39] Patrick Helber, Benjamin Bischke, Andreas R. Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12:2217–2226, 2019.

[40] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Lixuan Zhu, Samyak Parajuli, Mike Guo, Dawn Xiaodong Song, Jacob Steinhardt, and Justin Gilmer. The many faces of

robustness: A critical analysis of out-of-distribution generalization. In *IEEE International Conference on Computer Vision (ICCV)*, pages 8320–8329, 2021.

[41] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Xiaodong Song. Natural adversarial examples. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15257–15266, 2021.

[42] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations (ICLR)*, 2019.

[43] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981.

[44] Shuaiyi Huang, Qiuyue Wang, Songyang Zhang, Shipeng Yan, and Xuming He. Dynamic context correspondence network for semantic alignment. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2010–2019, 2019.

[45] Minyoung Huh, Pulkit Agrawal, and Alexei A Efros. What makes imagenet good for transfer learning? *ArXiv*, abs/1608.08614, 2016.

[46] Tak-Wai Hui, Xiaoou Tang, and Chen Change Loy. Liteflownet: A lightweight convolutional neural network for optical flow estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[47] Junhwa Hur, Hwasup Lim, Changsoo Park, and Sang Chul Ahn. Generalized deformable spatial pyramid: Geometry-preserving dense correspondence estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1392–1400, 2015.

[48] Junhwa Hur and Stefan Roth. Mirrorflow: Exploiting symmetries in joint optical flow and occlusion estimation. In *IEEE International Conference on Computer Vision (ICCV)*, pages 312–321, 2017.

[49] Junhwa Hur and Stefan Roth. Iterative residual refinement for joint optical flow and occlusion estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[50] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[51] Allan Jabri, Andrew Owens, and Alexei A. Efros. Space-time correspondence as a contrastive random walk. In *Neural Information Processing Systems (NeurIPS)*, 2020.

[52] Joel Janai, Fatma Güney, Anurag Ranjan, Michael J. Black, and Andreas Geiger. Unsupervised learning of multi-frame optical flow with occlusions. In *European Conference on Computer Vision (ECCV)*, 2018.

[53] J Yu Jason, Adam W Harley, and Konstantinos G Derpanis. Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness. In *European Conference on Computer Vision (ECCV)*. Springer, 2016.

[54] Sangryul Jeon, Seungryong Kim, D. Min, and K. Sohn. Parn: Pyramidal affine regression networks for dense semantic correspondence. In *European Conference on Computer Vision (ECCV)*, 2018.

[55] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning (ICML)*, pages 4904–4916, 2021.

[56] Ian T Jolliffe. Principal components in regression analysis. In *Principal component analysis*, pages 129–155. Springer, 1986.

[57] A. Kanazawa, D. Jacobs, and Manmohan Chandraker. Warpnet: Weakly supervised matching for single-view reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3253–3261, 2016.

[58] Guoliang Kang, Yunchao Wei, Yi Yang, Yueting Zhuang, and Alexander G Hauptmann. Pixel-level cycle association: A new perspective for domain adaptive semantic segmentation. In *Neural Information Processing Systems (NeurIPS)*, 2020.

[59] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. End-to-end learning of geometry and context for deep stereo regression. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[60] Jaechul Kim, Ce Liu, Fei Sha, and Kristen Grauman. Deformable spatial pyramid matching for fast dense correspondences. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2307–2314, 2013.

[61] Seungryong Kim, Stephen Lin, Sang Ryul Jeon, Dongbo Min, and Kwanghoon Sohn. Recurrent transformer networks for semantic correspondence. In *Neural Information Processing Systems (NeurIPS)*, 2018.

[62] Seungryong Kim, Dongbo Min, Bumsub Ham, Sangryul Jeon, Stephen Lin, and Kwanghoon Sohn. Fcss: Fully convolutional self-similarity for dense semantic correspondence. *IEEE Transactions on Pattern Recognition and Machine Intelligence (PAMI)*, 41:581–595, 2019.

[63] Seungryong Kim, Dongbo Min, Stephen Lin, and Kwanghoon Sohn. Dctm: Discrete-continuous transformation matching for semantic flow. In *IEEE International Conference on Computer Vision (ICCV)*, pages 4529–4538, 2017.

[64] Benjamin Klein, Guy Lev, Gil Sadeh, and Lior Wolf. Fisher vectors derived from hybrid gaussian-laplacian mixture models for image annotation. *ArXiv*, abs/1411.7399, 2014.

[65] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 554–561, 2013.

[66] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[67] Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

[68] Juho Lee, Yoonho Lee, Jungtaek Kim, Adam R. Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International Conference on Machine Learning (ICML)*, 2019.

[69] Junghyup Lee, Dohyung Kim, Jean Ponce, and Bumsub Ham. Sfnet: Learning object-aware semantic correspondence. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2273–2282, 2019.

[70] Chengkun Li, Haotian Liu, Liunian Harold Li, Pengchuan Zhang, Jyoti Aneja, Jianwei Yang, Ping Jin, Yong Jae Lee, Houdong Hu, Zicheng Liu, and Jianfeng Gao. Elevater: A benchmark and toolkit for evaluating language-augmented visual models. In *Neural Information Processing Systems (NeurIPS)*, 2022.

[71] Xueting Li, Sifei Liu, Shalini De Mello, Xiaolong Wang, Jan Kautz, and Ming-Hsuan Yang. Joint-task self-supervised learning for temporal correspondence. In *Neural Information Processing Systems (NeurIPS)*, 2019.

[72] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Universal style transfer via feature transforms. In *Neural Information Processing Systems (NeurIPS)*, 2017.

[73] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Flow-grounded spatial-temporal video prediction from still images. In *European Conference on Computer Vision (ECCV)*, 2018.

[74] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[75] Ce Liu, Jenny Yuen, and Antonio Torralba. Sift flow: Dense correspondence across scenes and its applications. *IEEE Transactions on Pattern Recognition and Machine Intelligence (PAMI)*, 33:978–994, 2011.

[76] Pengpeng Liu, Irwin King, Michael R. Lyu, and Jia Xu. Ddflow: Learning optical flow with unlabeled data distillation. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2019.

[77] Pengpeng Liu, Michael R. Lyu, Irwin King, and Jia Xu. Selflow: Self-supervised learning of optical flow. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[78] Yanbin Liu, Linchao Zhu, Makoto Yamada, and Yi Yang. Semantic correspondence as an optimal transport problem. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4462–4471, 2020.

[79] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal on Computer Vision (IJCV)*, 2004.

[80] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew B. Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *ArXiv*, abs/1306.5151, 2013.

[81] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[82] Simon Meister, Junhwa Hur, and Stefan Roth. Unflow: Unsupervised learning of optical flow with a bidirectional census loss. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2017.

[83] Moritz Menze, Christian Heipke, and Andreas Geiger. Joint 3d estimation of vehicles and scene flow. In *ISPRS Workshop on Image Sequence Analysis (ISA)*, 2015.

[84] Moritz Menze, Christian Heipke, and Andreas Geiger. Object scene flow. *ISPRS Journal of Photogrammetry and Remote Sensing (JPRS)*, 2018.

[85] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

[86] Juhong Min, Jongmin Lee, J. Ponce, and Minsu Cho. Hyperpixel flow: Semantic correspondence with multi-layer neural features. In *IEEE International Conference on Computer Vision (ICCV)*, pages 3394–3403, 2019.

[87] Juhong Min, Jongmin Lee, J. Ponce, and Minsu Cho. Learning to compose hypercolumns for visual correspondence. In *European Conference on Computer Vision (ECCV)*, 2020.

[88] Juhong Min, Jongmin Lee, Jean Ponce, and Minsu Cho. Spair-71k: A large-scale benchmark for semantic correspondence. *arXiv preprint arXiv:1908.10543*, 2019.

[89] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: Unsupervised learning using temporal order verification. In *European Conference on Computer Vision (ECCV)*, 2016.

[90] James Munkres. Algorithms for the assignment and transportation problems. *Journal of The Society for Industrial and Applied Mathematics*, 10:196–210, 1957.

[91] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP)*, pages 722–729, 2008.

[92] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision (ECCV)*, 2016.

[93] David Novotny, Diane Larlus, and Andrea Vedaldi. Anchornet: A weakly supervised network to learn geometry-sensitive features for semantic matching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5277–5286, 2017.

[94] Pedro O O Pinheiro, Amjad Almahairi, Ryan Benmalek, Florian Golemo, and Aaron C Courville. Unsupervised learning of dense visual representations. In *Neural Information Processing Systems (NeurIPS)*, volume 33, pages 4489–4500, 2020.

[95] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *ArXiv*, abs/1807.03748, 2018.

[96] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3498–3505, 2012.

[97] Deepak Pathak, Ross B. Girshick, Piotr Dollár, Trevor Darrell, and Bharath Hariharan. Learning features by watching objects move. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6024–6033, 2017.

[98] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, 2021.

[99] Anurag Ranjan and Michael J Black. Optical flow estimation using a spatial pyramid network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[100] Anurag Ranjan, Joel Janai, Andreas Geiger, and Michael J Black. Attacking optical flow. In *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[101] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning (ICML)*, 2019.

[102] Zhe Ren, Junchi Yan, Bingbing Ni, Bin Liu, Xiaokang Yang, and Hongyuan Zha. Unsupervised deep learning for optical flow estimation. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2017.

[103] Ignacio Rocco, R. Arandjelović, and Josef Sivic. Convolutional neural network architecture for geometric matching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 39–48, 2017.

[104] Ignacio Rocco, R. Arandjelović, and Josef Sivic. End-to-end weakly-supervised semantic alignment. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6917–6925, 2018.

[105] Ignacio Rocco, Mircea Cimpoi, Relja Arandjelović, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Neighbourhood consensus networks. In *Neural Information Processing Systems (NeurIPS)*, volume abs/1810.10510, 2018.

[106] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal on Computer Vision (IJCV)*, 47(1-3):7–42, 2002.

[107] Paul Hongsuck Seo, Jongmin Lee, Deunsol Jung, Bohyung Han, and Minsu Cho. Attentive semantic alignment with offset-aware correlation kernels. In *European Conference on Computer Vision (ECCV)*, 2018.

[108] Noam Shazeer and Mitchell Stern. Adafactor: Adaptive learning rates with sublinear memory cost. In *International Conference on Machine Learning (ICML)*, pages 4596–4604. PMLR, 2018.

[109] Jonathan Shen, Patrick Nguyen, Yonghui Wu, Zhifeng Chen, Mia X Chen, Ye Jia, Anjuli Kannan, Tara Sainath, Yuan Cao, Chung-Cheng Chiu, et al. Lingvo: a modular and scalable framework for sequence-to-sequence modeling. *ArXiv*, abs/1902.08295, 2019.

[110] Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. Flava: A foundational language and vision alignment model. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15638–15650, 2022.

[111] Richard Sinkhorn. Diagonal equivalence to matrices with prescribed row and column sums. *American Mathematical Monthly*, 74:402, 1967.

[112] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[113] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Models matter, so does training: An empirical study of cnns for optical flow estimation. *IEEE Transactions on Pattern Recognition and Machine Intelligence (PAMI)*, 2019.

[114] Tatsunori Taniai, Sudipta N Sinha, and Yoichi Sato. Joint recovery of dense correspondence and cosegmentation in two images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4246–4255, 2016.

[115] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision (ECCV)*, 2020.

[116] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herv'e J'egou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning (ICML)*, 2021.

[117] Yi-Hsuan Tsai, Ming-Hsuan Yang, and Michael J Black. Video segmentation via object flow. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[118] Vishaal Udandarao, Ankush Gupta, and Samuel Albanie. Sus-x: Training-free name-only transfer of vision-language models. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2725–2736, 2023.

[119] Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. In *Neural Information Processing Systems (NeurIPS)*, 2016.

[120] Aäron Van Den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *International Conference on Machine Learning (ICML)*, 2016.

[121] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

[122] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *International Conference on Machine Learning (ICML)*, 2008.

[123] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3156–3164, 2015.

[124] Carl Vondrick, Abhinav Shrivastava, A. Fathi, S. Guadarrama, and Kevin Murphy. Tracking emerges by colorizing videos. In *ECCV*, 2018.

[125] Haohan Wang, Songwei Ge, Eric P. Xing, and Zachary Chase Lipton. Learning robust global representations by penalizing local predictive power. In *Neural Information Processing Systems (NeurIPS)*, 2019.

[126] Liwei Wang, Yin Li, Jing Huang, and Svetlana Lazebnik. Learning two-branch neural networks for image-text matching tasks. *IEEE Transactions on Pattern Recognition and Machine Intelligence (PAMI)*, 41(2):394–407, 2018.

[127] Xiaolong Wang and Abhinav Gupta. Unsupervised learning of visual representations using videos. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2794–2802, 2015.

[128] Xiaolong Wang, Allan Jabri, and Alexei A. Efros. Learning correspondence from the cycle-consistency of time. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2561–2571, 2019.

[129] Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense contrastive learning for self-supervised visual pre-training. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[130] Yang Wang, Yezhou Yang, Zhenheng Yang, Liang Zhao, and Wei Xu. Occlusion aware unsupervised learning of optical flow. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[131] Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. Simvlm: Simple visual language model pretraining with weak supervision. In *International Conference on Learning Representations (ICLR)*. OpenReview.net, 2022.

[132] Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, et al. Robust fine-tuning of zero-shot models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7959–7971, 2022.

[133] Taihong Xiao, Jiapeng Hong, and Jinwen Ma. Dna-gan: Learning disentangled representations from multi-attribute images. In *International Conference on Learning Representations Workshop (ICLRW)* , 2018.

[134] Taihong Xiao, Jiapeng Hong, and Jinwen Ma. Elegant: Exchanging latent encodings with gan for transferring multiple face attributes. In *European Conference on Computer Vision (ECCV)*, pages 172–187, 2018.

[135] Zhenda Xie, Yutong Lin, Zheng Zhang, Yue Cao, Stephen Lin, and Han Hu. Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16684–16693, 2021.

[136] Jia Xu, René Ranftl, and Vladlen Koltun. Accurate Optical Flow via Direct Cost Volume Processing. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[137] Gengshan Yang and Deva Ramanan. Volumetric correspondence networks for optical flow. In *Neural Information Processing Systems (NeurIPS)*, 2019.

[138] Hongsheng Yang, Wen-Yan Lin, and Jiangbo Lu. Daisy filter flow: A generalized discrete approach to dense correspondences. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3406–3413, 2014.

[139] Zhichao Yin, Trevor Darrell, and Fisher Yu. Hierarchical discrete distribution decomposition for match density estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[140] Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[141] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *Transactions on Machine Learning Research (TMLR)*, 2022.

[142] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel C. F. Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, Ce Liu, Mengchen Liu, Zicheng Liu, Yumao Lu, Yu Shi, Lijuan Wang, Jianfeng Wang, Bin Xiao, Zhen Xiao, Jianwei Yang, Michael Zeng, Luowei Zhou, and Pengchuan Zhang. Florence: A new foundation model for computer vision. *ArXiv*, abs/2111.11432, 2021.

[143] Jure Zbontar and Yann LeCun. Stereo matching by training a convolutional neural network to compare image patches. *Journal of Machine Learning Research (JMLR)*, 17, 2016.

[144] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[145] Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. Lit: Zero-shot transfer with locked-image text tuning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18123–18133, 2022.

[146] Renrui Zhang, Rongyao Fang, Peng Gao, Wei Zhang, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free clip-adapter for better vision-language modeling. In *European Conference on Computer Vision (ECCV)*, 2022.

[147] Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful image colorization. In *European Conference on Computer Vision (ECCV)*, 2016.

[148] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[149] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal on Computer Vision (IJCV)*, 130:2337–2348, 2022.

[150] Shuchang Zhou, Taihong Xiao, Yi Yang, Dieqiao Feng, Qinyao He, and Weiran He. Genegan: Learning object transfiguration and attribute subspace from unpaired data. In *British Machine Vision Conference (BMVC)*, 2017.

[151] Tinghui Zhou, Philipp Krähenbühl, Mathieu Aubry, Qixing Huang, and Alexei A. Efros. Learning dense correspondence via 3d-guided cycle consistency. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 117–126, 2016.

[152] Tinghui Zhou, Y. Lee, S. Yu, and Alexei A. Efros. Flowweb: Joint image set alignment by weaving consistent, pixel-wise correspondences. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1191–1200, 2015.

[153] Xiaowei Zhou, Menglong Zhu, and Kostas Daniilidis. Multi-image matching via fast alternating minimization. In *IEEE International Conference on Computer Vision (ICCV)*, pages 4032–4040, 2015.

[154] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[155] Xiangyang Zhu, Renrui Zhang, Bowei He, Aojun Zhou, Dong Wang, Bin Zhao, and Peng Gao. Not all features matter: Enhancing few-shot clip with adaptive prior refinement. *ArXiv*, abs/2304.01195, 2023.

[156] Yuliang Zou, Zelun Luo, and Jia-Bin Huang. Df-net: Unsupervised joint learning of depth and flow using cross-task consistency. In *European Conference on Computer Vision (ECCV)*, 2018.