Supervised Estimation of Dense Optical Flow

Rishabh Gupta

Supervised Estimation of Dense Optical flow

DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Technology
in
Computer Science

by

Rishabh Gupta
[Roll No: CS-1810]

under the guidance of

Prof. Bhabatosh Chanda
Professor
Electronics and Communication Sciences Unit

Indian Statistical Institute
Kolkata-700108, India

July 2020

To my family and my guide

CERTIFICATE

This is to certify that the dissertation entitled **"Supervised Estimation of Dense Optical Flow"** submitted by **Rishabh Gupta** to Indian Statistical Institute, Kolkata, in partial fulfillment for the award of the degree of **Master of Technology in Computer Science** is a bonafide record of work carried out by him under my supervision and guidance. The dissertation has fulfilled all the requirements as per the regulations of this institute and, in my opinion, has reached the standard needed for submission.

Bhabatosh Chanda
Professor,
Electronics and Communication Sciences Unit,
Indian Statistical Institute,
Kolkata-700108, INDIA.

Acknowledgments

I would like to show my highest gratitude to my advisor, *Prof. Bhabatosh Chanda*, Electronics and Communication Sciences Unit, Indian Statistical Institute, Kolkata, for his guidance and continuous support and encouragement. He has literally taught me how to do good research, and motivated me with great insights and innovative ideas.

I would also like to thank, *Ranjan Mondal*, PhD Student, Electronics and Communication Sciences Unit, Indian Statistical Institute Indian Statistical Institute, Kolkata, for his valuable suggestions and discussions.

My deepest thanks to all the teachers of Indian Statistical Institute, for their valuable suggestions and discussions which added an important dimension to my research work.

Finally, I am very much thankful to my parents and family for their everlasting supports.

Last but not the least, I would like to thank all of my friends for their help and support. I thank all those, whom I have missed out from the above list.

Rishabh Gupta
Indian Statistical Institute
Kolkata - 700108 , India.

Abstract

End-to-end trained Convolutional Neural Network (CNN) have significantly advanced the field of computer vision in recent years, particularly high-level vision problems, because of its strong non-linear fitting ability. In context of optical flow, obtaining dense, ground truth per-pixel for real scenes is difficult and thus rarely available. But CNN in recent years demonstrated that dense optical flow estimation can be cast as a learning problem. However, the state of the art with regard to the quality of the flow has still been defined by traditional methods. In this thesis, firstly, we used a compact but effective CNN model, called U-Net, which contains an encoder part and a decoder part and used benchmark datasets: MPI-Sintel, KITTI and Middlebury; for training and evaluation, in a supervised manner. Secondly, we used some traditional energy-based loss function for dense optical flow estimation. Thirdly, we used backward warping with bilinear interpolation to predict first image and build occlusion mask using ground truth flow. Experimental results show that our proposed method is at par with state-of-the-art supervised CNN methods.

Keywords: *Dense Optical flow, Backward Image warping, Occlusion Mask, Convolutional Neural Network (CNN), Energy-Based Loss Functions.*

Contents

Chapter 1

Introduction

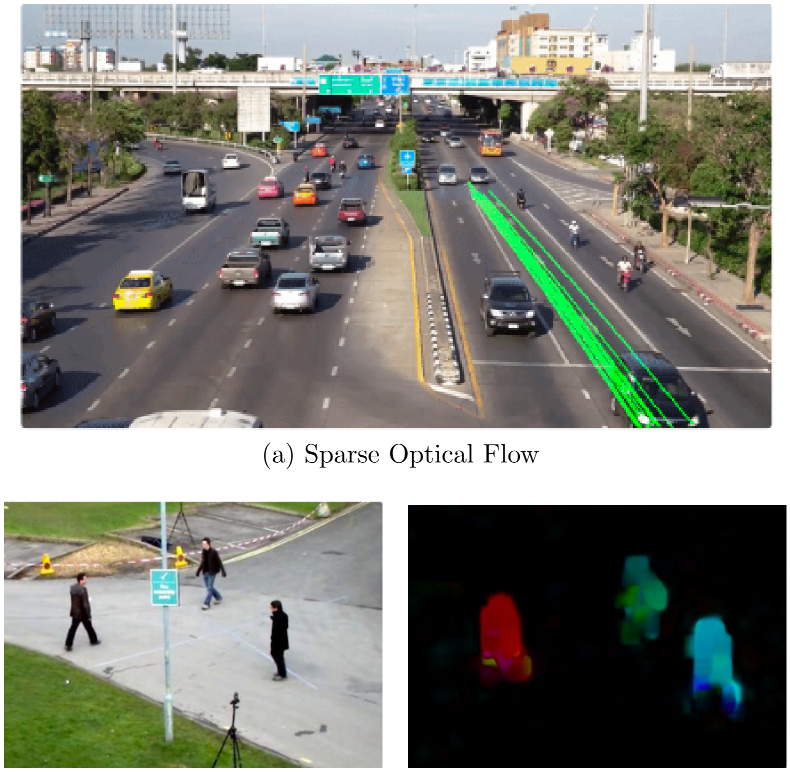
1.1 Introduction

Optical flow is the task of estimating per-pixel motion between video frames. More precisely, it describes a sparse or dense vector field, where each vector is the displacement vector assigned to a certain pixel position, to tell us where that pixel can be found in temporal space. The concept of optical flow was introduced by the American psychologist James J. Gibson in the 1940s, he used it to describe the visual stimulus provided to animals moving through the world. Optical flow can be divided into two type, depending on the density of pixel we are predicting it for.

- **Sparse Optical Flow:** Sparse optical flow gives the flow vectors of some "interesting features". Features in images are points of interest which present rich image content information such as corners and edges. These features correspondences is maintained from frame to frame and tracked to give optical flow.

- **Dense Optical Flow:** Dense optical flow attempts to compute the optical flow vector for every pixel position of each frame. It is complex and computationally slower than sparse optical estimation because it cannot be solved by finding feature correspondence just for few pixel position.

Optical flow has traditionally been approached as a handcrafted optimization problem. The most predominant way present in today's computer vision literature is introduced in the seminal work by Horn and Schunck [5] that minimize a global energy function consisting of a data and a smoothness term. Although the original Horn and Schunck model reveals many limitations, many of which have been tackled using addition or modifications to energy terms. For example, Motion discontinuities and occlusions can be estimated by employing non-quadratic penalizers in the smoothness term [13]. Violations of the constant brightness assumption can be considered



(a) Sparse Optical Flow

(b) Image for Dense Optical Flow

(c) Dense Optical Flow for (b)

Figure 1.1: (a) Sparse Optical Flow prediction superimposed over the scene.(b) First image frame from a sequence. (c) Dense Optical flow prediction for the first and its consecutive frame.

by using photometric invariant constraints, such as constancy of the gradient [35], higher order derivatives [27]. Such an approach has achieved considerable success.

1.1.1 Application

Vision Related Tasks where Optical flow is useful:

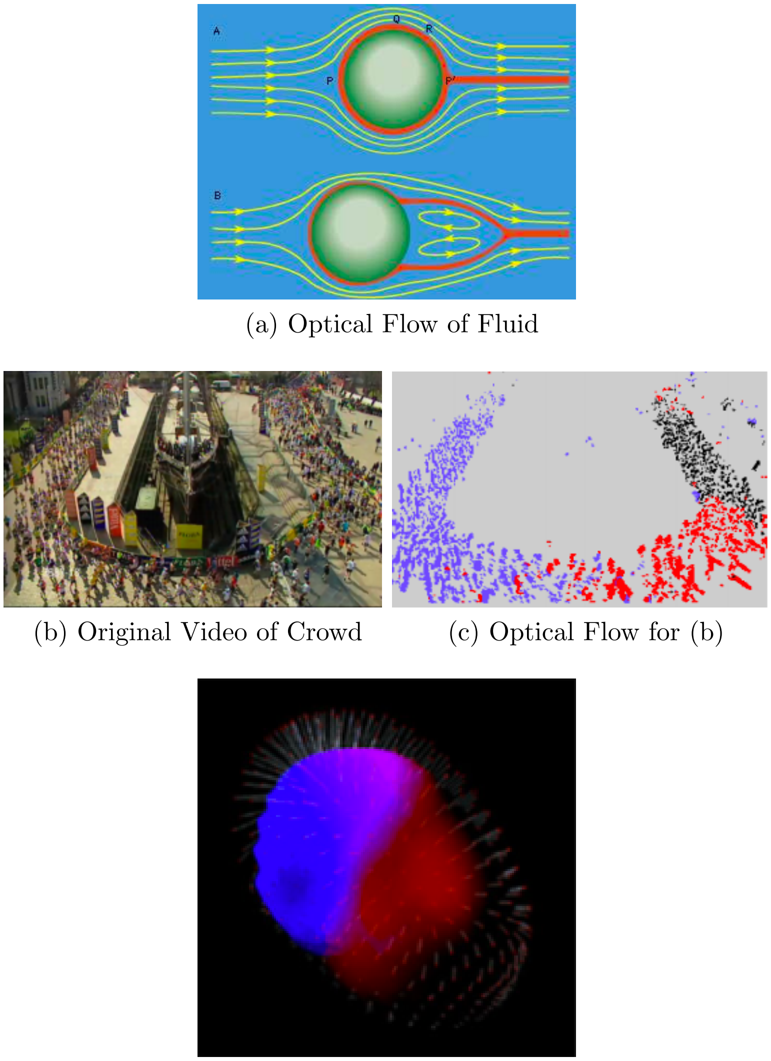
- **Action recognition:** The Temporal stream from video is used to recognize action from motion in the form of dense optical flow [18].
- **Learning Video Temporal Consistency**: Takes per frame processed videos with serious temporal flickering as inputs and generates temporally stable videos while maintaining perceptual similarity to the processed frames [25].
- **Ego-motion estimation**: The process of determining the position and orientation of a robot by analyzing the associated camera images. It has been used in

a wide variety of robotic applications, such as on the Mars Exploration Rovers [16].

- **Structure from motion(SfM)**: The camera parameters (intrinsic and extrinsic) need to be estimated jointly with the 3D structure while in Multi-View Stereo (MVS) [16].
- **Object Tracking**: To estimate the state (location, velocity and acceleration) of one or multiple objects over time given measurements of a sensor [14].
- **Image and Video Compression**: With the growing popularity of visual data usage, image and video coding is an active and dynamic field [32].
- **Region Segmentation within Images:** Discontinuities in the optical flow can help in segmenting images into regions that correspond to different objects.

Industrial or Business Domains for which Optical Flow is useful:

- **Security**: Eg: In Crowd Suvelliance Videos to detect abnormal crowd behavior :crowd spread behavior, crowd gather behavior and crowd collective movement behavior.
- **Robotics / Autonomous Driving Car**: Ego Motion Estimation of monocular camera mounted on an Autonomous Driving Car or Mars Exploration Rovers.
- **Daily functioning of persons with low vision**: Low vision [persons] often are unable to recognize their surroundings due to blurred images. So, motion-generated information helps them to perceive events in their surroundings
- **Aviation Industry**: Like in military helicopters, pilot use optical flow and other optical cues to identify objects and visualize terrain features
- **Medical Imaging**: Eg: applied to time series for the analysis of tumor growth; for denoising x-ray image sequences or for Cardiac Motion Estimation [19].
- **Entertainment Industry**: Eg: Innovative game controllers use motion trajectories of humans as input device
- **Automobile Manufacturing and Weather Forecasting**: One fluid is air; its flow fields are used both scientifically and commercially for weather forecasting and the optimization of car and airplane shapes
- **3D reconstruction**: Eg: Google Street View which uses motion and stereo vision to reconstruct entire cities in 3D.



(a) Optical Flow of Fluid

(b) Original Video of Crowd

(c) Optical Flow for (b)

(d) Optical flow for Cardiac Motion

Figure 1.2: Few Examples of Optical Flow usage in different domains.

1.2 Motivation and Problem Statement

As we saw earlier in section 1.1; optical flow can either be helpful directly ,such as in Video Compression, or can aid in other vision tasks, such as Object Tracking. But still traditional methods are at large from predicting ground truth dense optical flow. Firstly, due to data coming in real-time and in huge amount, predicting dense optical flow by optimizing complex and handcrafted energy-based function by processing two frames at a time, is computationally expensive and time consuming. So there is a need of a reusable framework that can predict dense optical flow instead of rebuilding energy function and optimizing it from scratch. Secondly, best systems are limited by difficulties including fast-moving objects, occlusions, motion blur,and textureless surface. Due to these difficulties any further progress appeared challenging. But in the recent decade, advancement in deep learning has shown us that optical flow prediction can be posed as a learning problem .Moreover, Convolutional Neural Network has given state-of-the-art results in many vision related tasks. So the motive of the thesis is to develop a Deep CNN model that can predict non-linear optical flow for complex scenes in real-time and tried to emphasize the fact that synergy between traditional approaches and deep learning can give us greater performance gains.

1.3 Thesis Outline

The rest of paper is organized as follows :

- Chapter 2: This Chapter reviews the recent methods on optical flow. It starts by briefly reviewing the traditional methods in use to success stories of convolutional neural network in optical flow prediction.
- Chapter 3: This chapter provides an outline of dataset used, preprocessing and augmentation done to make it robust for model training. it also highlights prerequisite such color-coding of flow vectors, concept of backward warping and building of occlusion mask.
- Chapter 4: This chapter mainly discuss about the proposed method. It engulfs architecture of our proposed network and loss function used for prediction and its meaning. And also shed light on training details.
- Chapter 5: This chapter discusses about the Hardware and software used for the thesis and and also gives a brief explanation for the Hyper-parameter tuning. In the end tells the results carried out on the test set using our trained model. Also through comparison and discussion brings out key factors and areas responsible for improvements or worsening in our prediction, both quantitative and qualitative.
- Chapter 6: Finally, this chapter gives a brief conclusion on the observed result from our experiment and what are the areas of improvement we see and continuing along the line what we propose to be doing next.
- The document ends with the Bibliography and reference materials.

Chapter 2

Related Work

2.1 Traditional Energy-Based Optical Flow Method

A variational optical flow method; proposed by Horn eat al. in [5]; have played a dominant role in optical flow estimation by coupling the brightness constancy and spatial smoothness assumptions using an energy function. From then ,a lot of work followed the classical energy function framework. Sun et al. [10] while discovering the secrets of how the classical energy function and their optimization methods, finds that the median filtering of intermediate flow fields during optimization increases performance gains and introduces a non-local term that robustly integrates flow estimates over large spatial neighborhoods. Black and Anandan [22] introduce a robust framework to deal with outliers, i.e., brightness inconstancy and spatial discontinuities. As it is computationally impractical to perform a full search, a coarse-to- fine, warping-based approach is adopted, Bruhn et al. [1]. While warping schemes work well in all cases where the small structures move more or less the same way as larger scale structures, the approach failed as soon as the relative motion of a small scale structure is larger than its own scale. To address this large displacement problem Brox et al. [35] embed descriptor matching into the variational framework. Descriptor matching has its own drawbacks, it can give false matches, ambiguous matches, or has low precision, which is further improved by follow-up methods [30], [20]. Bailer et al. [7] present a dense correspondence field approach for optical flow estimation, which uses a novel hierarchical correspondence field search strategy to match descriptor. The variational approach is the most popular framework for optical flow. However, it requires making use of the prior assumptions which are defined by human and deviate from the reality. Secondly, solving complex optimization problems and is computationally expensive for real-time applications. Moreover, these cannot automatically learn from a large amount of data to obtain a model that can generate optical flow end-to-end, and need to be pre-defined. Hence, most of recent works focus on deep learning and use convolutional neural network for learning optical flow.

2.2 Deep Learning-Based Optical Flow Method

Motivated by the success of deep learning techniques in matching or correspondence estimation problems paved ways for predicting optical flow estimation. Andreas et al. [12] uses deep nets to extract context-aware features to compute optical flow through patch matching by comparing each pixel in the reference image to every pixel in the target image. Bai et al. [21] further extended the patch matching based methods to segmentwise epipolar flow. Later, Bailer et al.[8] proposes a robust thresholded hinge loss for Siamese networks to learn CNN-based patch matching features for different image scales. Jia Xu et al. [17] accelerated the process of patch by exploiting the regularity in cost volume, thus obtaining optical flow results with high accuracy and fast speed. Meanwhile U-Net architecture, using CNN, was proposed.

2.2.1 U-Net based Optical Flow Estimation

Taking advantage of U-Net architecture [28], Dosovitskiy et al. [2], proposed two networks, FlowNetS and FlowNetC for learning optical flow end-to-end, which take two consecutive input images and output a dense optical flow map using an encoder-decoder architecture.Following which many networks for learning optical flow are proposed. Lopez et al. [38] combine U-Net with coarse-and-fine reasoning i.e. it combines a coarse result based on the solution of pixel-wise horizontal and vertical classification problems with a fine one obtained through regression predictions and learn optical flow with supervised manner. Xiang et al. [40] re-emphasizing the variational approach for optical flow estimation, propose a novel loss function which combines prior assumptions with supervised loss term and implement it on FlowNet. To obtain more refined flow fields, FlowNet2.0 [11] stacked several U-Net to form a large network and also included warping of the second image at each intermediate optical flow for iterative refinement. Although the stacking operation can improve the accuracy of flow estimation, the training process is complex and the sub-networks need to be trained one-by-one. Ranjan and Black [4] in addition to several U-Nets stacked together, combined a classical spatial-pyramid formulation , called SpyNet. This allowed to estimates large motions in a coarse-to-fine approach by warping one image of a pair at each pyramid level by the current flow estimate and computing an update to the flow. Based on [4], Hu et al. [29] further present a combination of recurrent spatial pyramid network(RecSPy) and the proposed energy-based refinement for learning optical flow. Hui et al. [37] propose a lightweight network for optical flow estimation, which uses pyramid network and feature warping to refine flow fields. The main advantage of spatial pyramid network is that the parameters of the model is less and the speed is fast. However, the accuracy of [4] and [29] is close to [2].

Since most deep networks are built to predict flow using two consecutive frames and trained with supervised learning, it would require a large amount of training data to obtain reasonably high accuracy. Unfortunately, most large-scale flow datasets are from synthetic movies and ground-truth motion labels in real world videos are generally hard to annotate. To overcome this problem, unsupervised learning framework is proposed to utilize the resources of unlabeled videos.However, the performance of the unsupervised methods still has a relatively large gap compared to their supervised counterparts.Ren et al [43] aim to learn optical flow with unsupervised manner, which usually employ well-proven prior constraints used in knowledge-driven approaches to guide the network training such as brightness constancy, gradient constancy and spatial smoothness constraints. Zhu and Newsam [42] introduce DenseNet into learning optical flow which can be viewed as a com-bination of dense block and U-Net. It provides shortcut connections throughout the network, which leads to implicit deep supervision. UnFlow [33] further introduces the stacking architecture into unsupervised learning optical flow and uses a robust census loss function instead of using brightness loss. Its result on the KITTI dataset, outperforms previous unsupervised deep networks by a large margin, and is even more accurate than similar supervised methods.

2.2.2 Occlusion-Aware Optical Flow Estimation

Since occlusion is a consequence of depth and motion i.e. background depth information getting occluded by the moving foreground pixels, thus it becomes inevitable to model occlusion in order to accurately estimate flow.Most optical flow methods detect occlusion as outliers and predict target pixels in the occluded regions as a constant value or through interpolation . Andreas et al. [12] perform feature matching by comparing each pixel in the reference image to every pixel in the target image thus treating occluded pixels as outlier for optical flow prediction. Chen et al. [31] performs consistency checking on estimated forward and backward optical flow and identifying inconsistent matches in flow as occluded pixel position, are discarded. These occluded areas is then extrapolated with optical flow. Other methods incorporate occlusion estimation directly into the energy minimisation the best non-CNN method MirrorFlow [15] fully exploit the symmetry properties that characterize optical flow and occlusions - specifically; forward-backward consistency and occlusion-disocclusion symmetry in a single joint optimisation.

Most of the current state-of-the-art CNN networks do not explicitly deal with occlusions. The network in UnFlow [33] estimates the forward and backward flows independently and uses the forward-backward consistency check to estimate the occlusions. The estimated occlusions are then used for network training only. In LiteFlowNet [37] an occlusion probability map is a function of brightness inconsistency between the reference frame and warped target frame. The occlusion probability map is used in a flow regularisation module. In work by M.Neoral et al. [24] showed occlusions also facilitate multi-frame optical estimation. Using the occlusions and flow from the previous flow, the network has prior information about the motion to be used when no correspondences are available. In MaskFlowNet [34] propose an asymmetric occlusion aware feature matching module, which can learn a rough occlusion mask that filters useless (occluded) areas immediately after feature warping without any explicit supervision. The proposed module can be easily integrated into any end-to-end network architectures and can jointly estimate occlusions without any explicit supervision in a single forward pass.

Chapter 3

Dataset Description and Prerequisite

3.1 Dataset

In recent years , most work of the optical flow prediction are done mainly on these 3 datasets: Middlebury, KITTI and MPI-Sintel. We use consecutive RGB image pairs as input $X \in \mathbb{R}^{H \times W \times 3}$ from these pairs to predict corresponding dense optical flow $Y \in \mathbb{R}^{H \times W \times 2}$.

MPI-Sintel This data is derived from the animated short film Sintel [36] and is available on website [39]. It contains richly varied motion, illumination, scene structure, material properties, atmospheric effects, blur, etc. The Sintel flow data set provides 1628 frames of ground truth flow (100 times Middlebury) in separate test (564 frames, withheld) and training sets (1064 frames). Resolution of frames is 1024×436. It contains large motions ,as large as 100 pixels per frame, including small objects moving quickly. Dataset has *"Render Passes"* : *albedo*, *clean* and *final*; each pass adds complexity as illumination, shadowing effect, motion blur and more. Dataset set is also divided as sequences from movie with a frame rate of 24 frames per second and each sequence being 50 frames long, giving 49 flow fields per sequence.

Middleburry Most of the Middlebury sequences are 8 frames long, with several only being 2 frames long. Ground truth flow is provided only for one pair of frames in each sequence. Middlebury images range from 548×388 to 640×480 (plus the Yosemite sequence which is only 316×252). Middlebury motions are quite small (up to 12 pixels per frame in the real imagery and 35 pixels per frame in the synthetic) .

KITTI The KITTI dataset consists of real road scenes captured by a stereo camera mounted on a moving car and simultaneously a laser scanner provides accurate yet sparse optical flow ground truth for a small number of images, which make up the KITTI 2012 [3] and KITTI 2015 [23] flow benchmarks. In addition, a large dataset

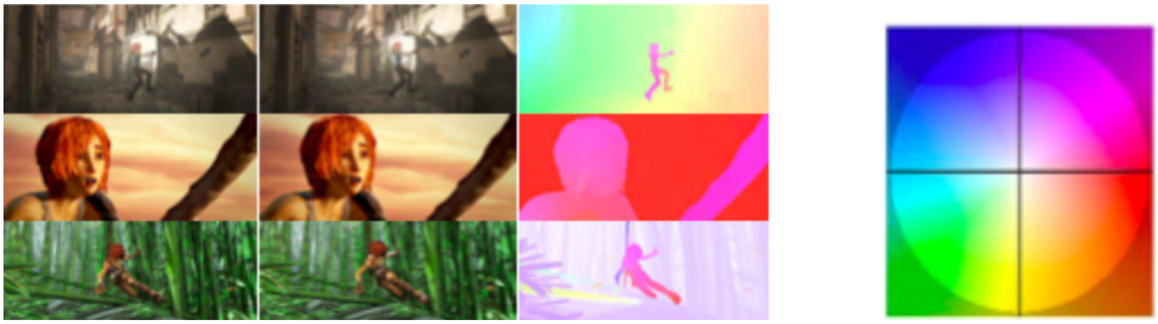


Figure 3.1: (Left) Example from MPI-Sintel Dataset: image1, image2, ground truth flow prediction. (Right) Coding Color of the flow vectors.

of raw 1392 × 512 image sequences is provided without ground truth. But we use the 394 image pairs with ground truth from the KITTI 2012 and KITTI 2015 training sets. KITTI 2012 has 195 testing and 194 training pairs and KITTI 2015 has 200 testing and 200 training pairs. In total we have 394 image pairs with ground truth from the KITTI 2012 and KITTI 2015 which we use as training set and rest are used for evaluation.

Pre-processing for training and evaluation When trained and tested separately for each dataset ;for Sintel and KITTI, no preprocessing is required as training and test set both have constant frame resolution; for middlebury we resize our images to 832 × 256. When we train on mixture of data; to prevent model from overfittting to single dataset; we shuffle and re-scale all images and flow to 832 × 256 using bilinear interpolation. Zero-padding the input images up to the next valid size introduces visible artifacts at the image boundaries and significantly increases the error. In addition we can do perturbation to our image sequences such as additive Gaussian noise ($0 < \sigma \leq 0.04$), random additive brightness changes, random horizontal and vertical flipping.

3.2 Pre-Requisite

3.2.1 Optical Flow Color Coding

For optical flow visualization we use the color coding of Butler et al.[6]. The color coding scheme is illustrated in Figure 3.2. Hue represents the direction of the displacement vector, while the intensity of the color represents its magnitude. White color corresponds to no motion. Because the range of motions is very different in different image sequences, we scale the flow fields before visualization: independently for each image pair shown in figures, and independently for each video fragment in the supplementary video. Scaling is always the same for all methods being compared.

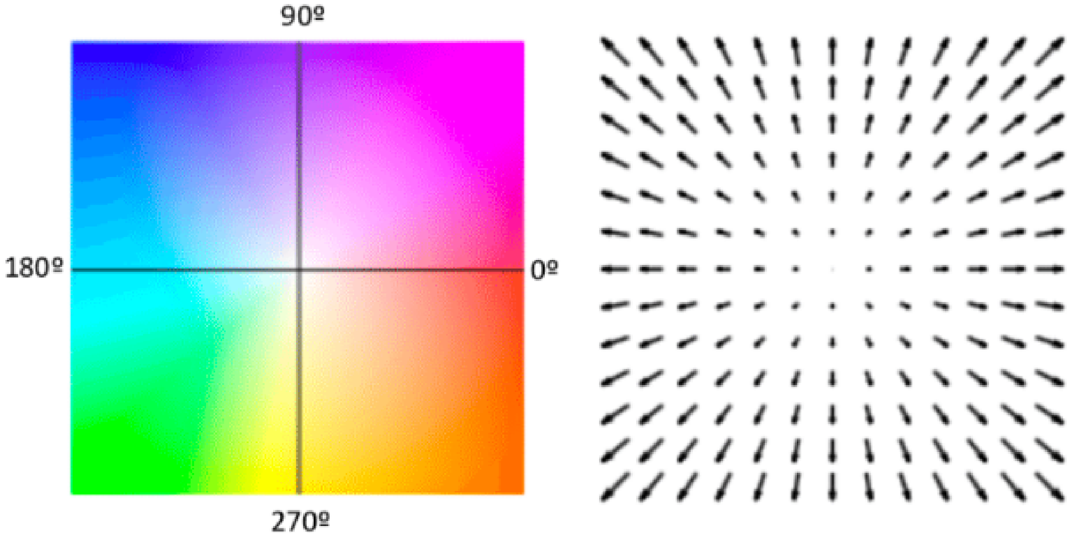


Figure 3.2: The visualization of flow fields. (Left Side): Color code visualization, and (Right Side): Arrow visualization

3.2.2 Backward Warping

The backward warping module is used to reconstruct I_1 from I_2 using the forward optical flow F_{12}. So if flow vector is a fractional number, which can happen due to re-scaling or resizing of ground truth optical flow or optical flow predicted by our model. So in that case, the warped pixel doesn't get translated to grid point of our image coordinate system. And if not addressed, most of warped pixel location will lose information and appear dark, which in turn will badly affect our loss by reflecting higher cost and thus increasing the difficulty in learning for our model .

More concretely, when we use the estimated optical flow F_{12} to warp I_2 back to reconstruct $\widetilde{I_1}$ at a grid point (x_1, y_1), we first translate the grid point (x_1, y_1) in I_1 (the yellow square in Figure 3.3) to (x_2, y_2)=$(x_1 + F^x_{12}(x_2, y_2), y_1 + F^y_{12}(x_1, y_1))$ in I_2, where F^x_{12} is flow field in horizontal direction and F^y_{12}. is flow field direction in vertical direction. Because the point (x_2, y_2) is not on the grid point in I_2, we need to do bilinear sampling to obtain its value and to do that we need to first obtain four nearest neighbour (the black dots in Figure 3.3) to translated (x_2, y_2), as:

$$(x_2^1, y_2^1) = (x_1 + \lfloor F^x_{12}(x_1, y_1) \rfloor, y_1 + \lfloor F^y_{12}(x_1, y_1) \rfloor) \tag{3.1}$$

$$(x_2^2, y_2^2) = (x_1 + \lfloor F^x_{12}(x_1, y_1) \rfloor, y_1 + \lceil F^y_{12}(x_1, y_1) \rceil) \tag{3.2}$$

$$(x_2^3, y_2^3) = (x_1 + \lceil F^x_{12}(x_1, y_1) \rceil , y_1 + \lfloor F^y_{12}(x_1, y_1) \rfloor) \tag{3.3}$$

$$(x_2^4, y_2^4) = (x_1 + \lceil F^x_{12}(x_1, y_1) \rceil , y_1 + \lceil F^y_{12}(x_1, y_1) \rceil) \tag{3.4}$$

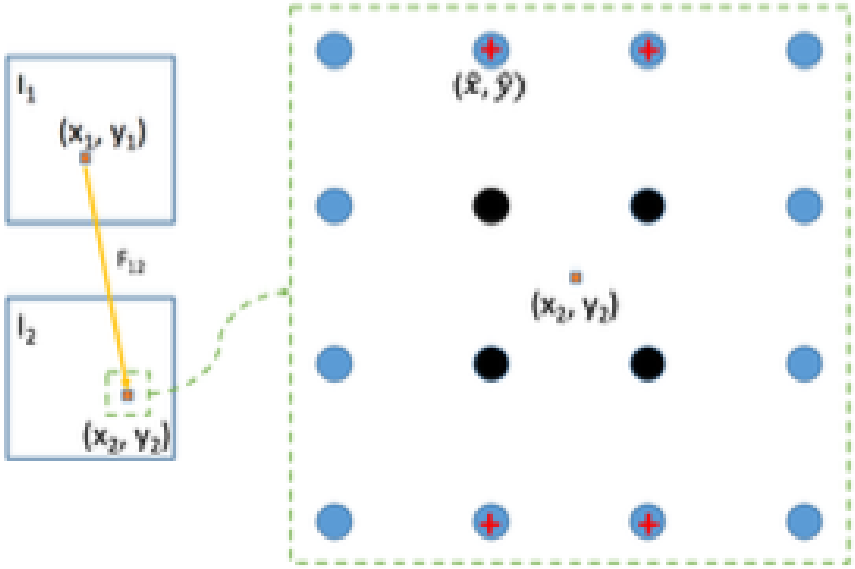


Figure 3.3: Backward Warping: The large green box on the right side is a zoom view of the small green box on the left side. Black dots are the 4 nearest neighbours.[41]

The above four location are of I_2. Now we need to obtain bi-linear weights as:

$$\theta_x = F^x_{12}(x_1, y_1) - \lfloor F^x_{12}(x_1, y_1) \rfloor, \bar{\theta_x} = 1 - \lfloor \theta_x \rfloor \tag{3.5}$$

$$\theta_y = F^x_{12}(x_1, y_1) - \lfloor F^y_{12}(x_1, y_1) \rfloor, \bar{\theta_y} = 1 - \lfloor \theta_y \rfloor \tag{3.6}$$

Now, the reconstructed $\widetilde{I_1}$ at (x_1, y_1) using bi-linear interpolation of the four nearest neighbour of I_2 i.e. (3.1), (3.2), (3.3) and (3.4) and corresponding product bilinear weights from (3.5) and (3.6) is:

$$\widetilde{\mathbf{I_1}}(\mathbf{x_1}, \mathbf{y_1}) = \bar{\theta}_{\mathbf{x}}\bar{\theta}_{\mathbf{y}}\mathbf{I_2}(\mathbf{x_2^1}, \mathbf{y_2^1}) + \bar{\theta}_{\mathbf{x}}\theta_{\mathbf{y}}\mathbf{I_2}(\mathbf{x_2^2}, \mathbf{y_2^2}) + \theta_{\mathbf{x}}\bar{\theta}_{\mathbf{y}}\mathbf{I_2}(\mathbf{x_2^3}, \mathbf{y_2^3}) + \theta_{\mathbf{x}}\theta_{\mathbf{y}}\mathbf{I_2}(\mathbf{x_2^4}, \mathbf{y_2^4}) \tag{3.7}$$

3.2.3 Occlusion mask for Backward Warping using ground truth optical flow

Here we will consider two types of occlusions o calculate occluded region, since pixels in these locations violates our brightness/photometric constancy and gradient constancy assumption and could limit the optical flow estimation accuracy since the loss

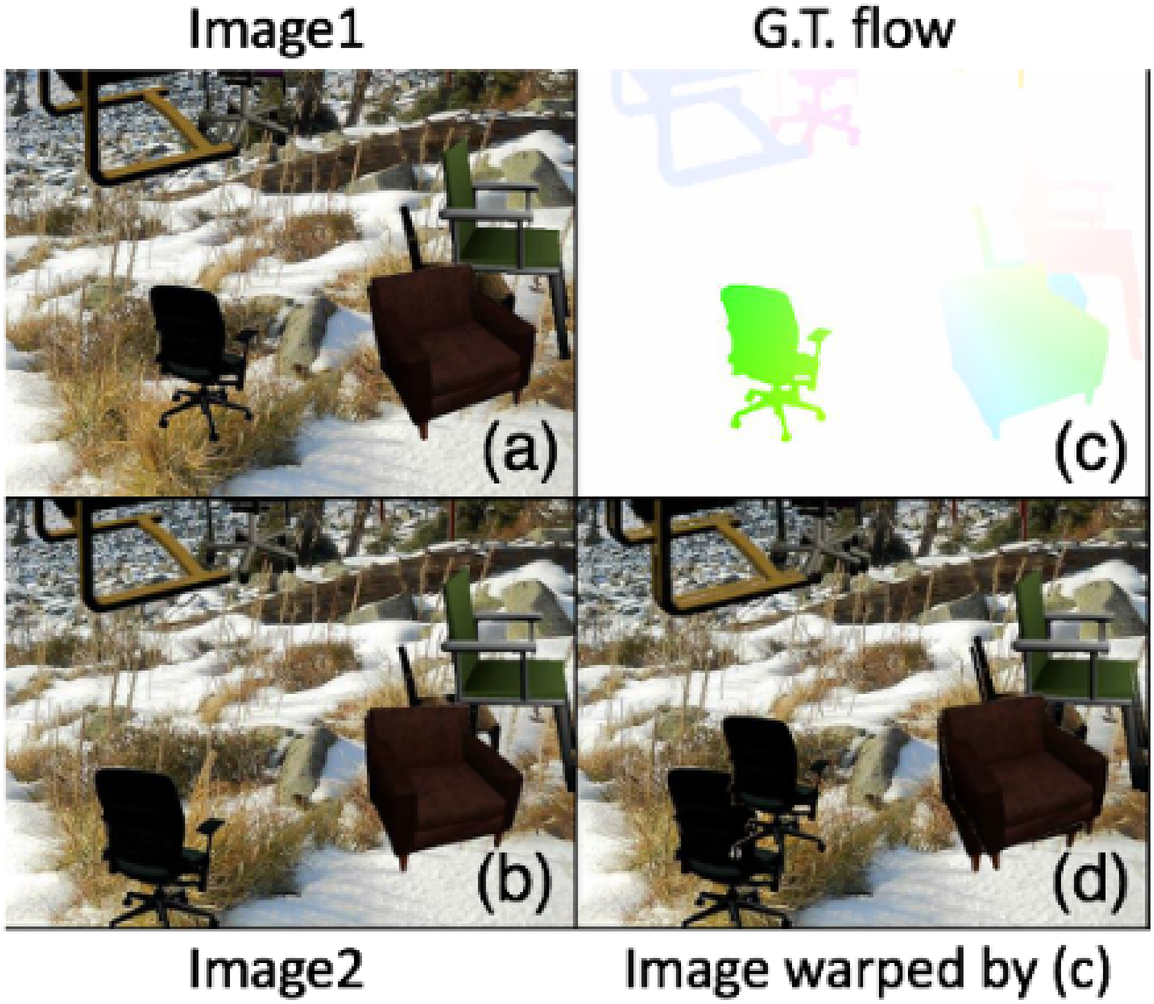


Figure 3.4: (a) Input frame 1. (b) Input frame 2. (c) Ground-truth optical flow. (d) Image warped by ground-truth optical flow

function would prefer to compensate the occluded regions by moving other pixels. They are:

- We will firstly consider those pixel position for occlusion, which on warping the image I_1 using flow field do not lie on grid points in I_2. For doing so, we translate grid point (x_1, y_1) in I_1 to (x_2, y_2)=$(x_1 + \lfloor F_{12}^x(x_1, y_1) \rfloor, y_1 + \lfloor F_{12}^y(x_1, y_1) \rfloor)$, and then if this translated position lies outside our image boundary, then it is occluded.

- Secondly, during backward warping, by using I_2 and F_{12}(ground truth forward optical flow) in equation (3.7), to reconstruct I_1. We do not get I_1, instead we get $\widetilde{I_1}$; this is due to fraction of pixel positions in I_1 which remain occluded in

$\widetilde{I}_1$. For example, in Figure 3.4, when we backward warp Image2 (Figure 3.4b) using ground truth (Figure 3.4c), then the resulting image (Figure 3.4d) has two chairs in it. Out of the two chairs, the chair on the top-right is the real chair while the bottom-left chair occlude some background of Image1. So the bottom-left chair pixel positions in both Image1 and Image2 are inconsistent for photometric comparison and hence contribute to occluded region.

For learning occlusion mask in supervised method , we use ground truth dense optical flow F_{12} to find occluded region. We first create and initialize all mask values to zero. Then for the two type of occlusion, occlusion mask is calculated simultaneously using translated point (x_2, y_2). Given as,

$$\text{Initially, } O(i, j) = 0, \forall (i, j) \in \mathbb{R}^{\mathbb{W} \times \mathbb{H}}$$

For calculating our first type of occlusion, we calculate

$$(x_2, y_2) = (x_1 + \lfloor F_{12}^x(x_1, y_1) \rfloor, y_1 + \lfloor F_{12}^y(x_1, y_1) \rfloor), \forall (x_1, y_1) \in \mathbb{R}^{\mathbb{W} \times \mathbb{H}}$$

. Then at (x_1, y_1) our occlusion is:

$$\mathbf{O(x_1, y_1)} = \begin{cases} 1, & \text{if } (x_2 \geq W) \text{ OR } (x_2 \leq 0) \\ 1, & \text{if } (y_2 \geq H) \text{ OR } (y_2 \leq 0) \\ 0, & \text{otherwise} \end{cases} \tag{3.8}$$

where; W is width of Image frame and H is height of image frame.
For our second type of occlusion, we extend it on previously calculted mask and (x_2, y_2). We check flow, $F_{12}(x_2, y_2)$ and give occlusion as:

$$\mathbf{O(x_2, y_2)} = \begin{cases} 1, & \text{if } F_{12}(x_2, y_2) = (0, 0) \;\; \& \;\; (x_2, y_2) \neq (x_1, y_1) \\ 0, & \text{otherwise} \end{cases} \tag{3.9}$$

Equation (3.9) is our final mask with $\mathbf{O(x,y) = 1}$; signifying occluded pixel and $\mathbf{O(x,y) = 0}$; signifying non-occluded pixel position.

Chapter 4

Proposed Method

4.1 Architecture

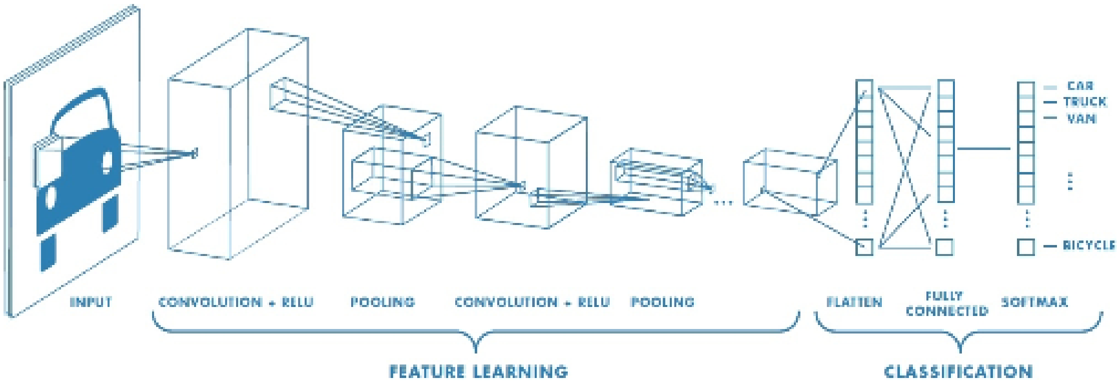


Figure 4.1

As seen in Figure 4.1, Convolutional Neutral Network's (or CNN's), just like other neural network, consists of stacks of layers. Commonly used layers are:

- **Convolution Layer** puts high dimensional inputs, such as images, through a set of convolution filter applied in sliding-window fashion to activate certain input features. Also promotes local connectivity and parameter sharing.

- **Pooling Layer** performs nonlinear down-sampling, there are a number of nonlinear functions to implement pooling. Intuition is to learn rough location of feature relative to other feature rather than exact location of the feature.

- **Rectified linear unit (ReLU)** allows for faster and more effective training by mapping negative values to zero and maintaining positive values. This is

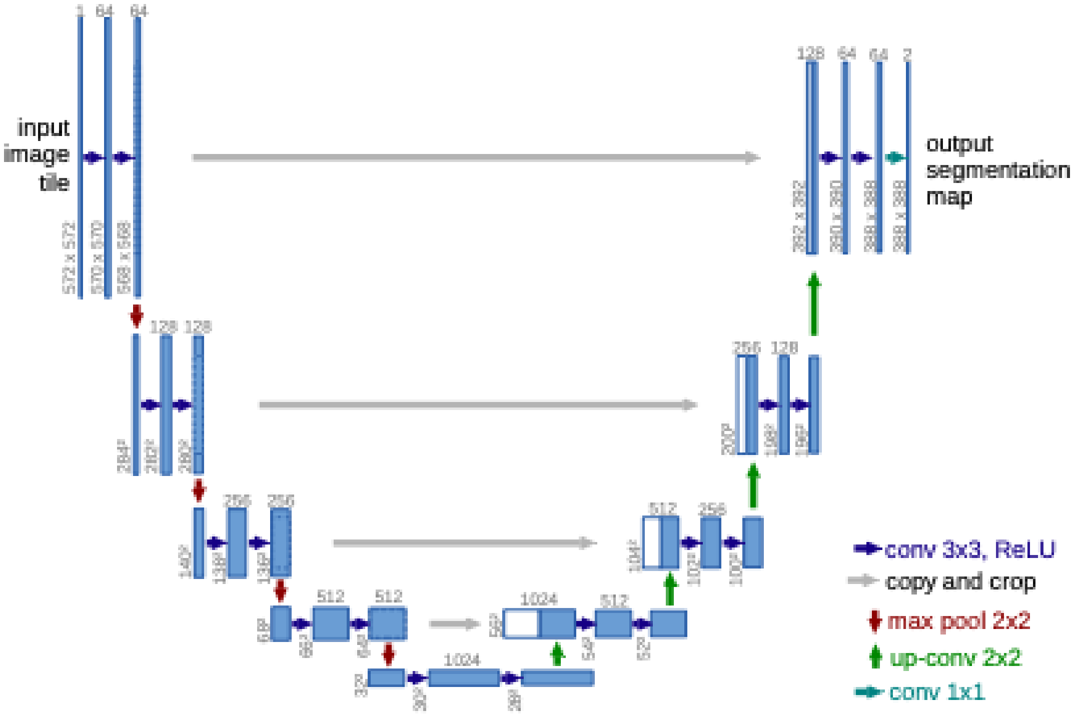


Figure 4.2: Similar to our U-Net network (An encoder-decoder architecture).

sometimes referred to as activation, because only the activated features are carried forward into the next layer.

- **Fully connected layer** serves to bring out the meaning from the features of the last convolution layer, i.e. how the presence of a certain feature related to all the other features .Neurons in a fully connected layer have connections to all features in the previous layer.

- The final layer of the CNN architecture uses a classification layer such as softmax to provide the classification output.

In our thesis, inspired by Ronneberger et al. [28] we proposed to use U-Net architecture, as illustrated in Figure 4.2, but with few differences. Firstly, at input layer we stack two consecutive image to get input of size $\mathbb{R}^{H \times W \times 6}$. The network has same two parts, encoder and decoder architecture, but instead of 4 level we used 3 level of encoding and decoding. The encoder part contains a series of convolution layers, here instead of two 3x3 unpadded convolutions, we used three 3x3 convolution with same padding. After each convolution layer instead of ReLU, we used Tanh activation layer and do batch normalization, for facilitating faster learning using gradient descent. To down-sample instead of max-pooling we used ”down-convolution”, convolution with a stride of 2 × 2 and doubled the number of feature channels. Our feature channel varies in (16,32,64). At every level in the decoder part, consists of an upsampling of the feature map followed by a 2x2 convolution ("up-convolution") that halves the number of feature channels, a concatenation with the corresponding feature map from the encoding part of network without any cropping, and more than three 3x3 convolutions, each followed by Tanh. At the final layer of predicting optical flow 3x3 convolution is used with linear activation function. In total the network has 29 convolutional layers instead of 23. And total parameters to learn is around 0.2 million.

4.2 Loss Function

The training set is comprised of pairs of temporally consecutive images, $I_1(x, y)$ and $I_2(x, y)$, and ground truth optical flow $F_{12}(x, y)$ and the flow predicted by our model as $\widetilde{F_{12}}(x, y)$. Now we calculate our backward warped image, $\widetilde{I_1}(x, y)$, using $I_2(x, y)$ and $\widetilde{F_{12}}(x, y)$ as input to our equation (3.7). And will using our ground truth flow ,$F_{12}(x, y)$, for computing occlusion mask $O(x, y)$ in equation (3.9).
Our first loss is based on brightness constancy assumption over non-occluded pixels is defined as:

$$L_b = \sum\nolimits_{(x,y)\in\mathbb{R}^{\mathbb{W}\times\mathbb{H}}} (1 - O(x, y)) \times \rho(\left\| I_1(x, y) - \widetilde{I_1}(x, y) \right\|^2) \tag{4.1}$$

where $\|\cdot\|$ is Euclidean Norm or L2 norm of vectors and $\rho(x) = (x^2 + \epsilon^2)^\gamma$ is the robust generalized Charbonnier penalty function with $\gamma = 0.45$ and $\epsilon = 0.01$.
The brightness constancy assumption has an obvious drawback which is quite susceptible to the slight changes in brightness. To address this issue, gradient constancy assumption is employed in many traditional methods for optical flow estimation, which can be given as following:

$$L_g = \sum\nolimits_{(x,y)\in\mathbb{R}^{\mathbb{W}\times\mathbb{H}}} (1 - O(x, y)) \times \rho(\left\| \nabla I_1(x, y) - \nabla\widetilde{I_1}(x, y) \right\|^2) \tag{4.2}$$

where $\nabla = (\partial_x, \partial_y)^T$ denotes spatial gradient taken in x and y direction.

The above two losses takes care of brightness consistency between the input and the warped image. To make flow prediction robust, we will also include standard L2 loss between our ground truth flow and predicted flow, also known as *Endpoint Error(EPE)*. This loss is calculated over all pixels, occluded as well as non-occluded, and given as:

$$L_{epe} = \sum\nolimits_{(x,y)\in\mathbb{R}^{\mathbb{W}\times\mathbb{H}}} \rho(\left\| F_{12}(x, y) - \widetilde{F_{12}}(x, y) \right\|^2) \tag{4.3}$$

The L_{epe} helps in estimating the optical flow without taking any interaction between neighbor pixels into account. Hence, it is useful to introduce the smoothness assumption of the flow field, it not only assumes neighboring points on the object have similar velocities and also velocities vary smoothly almost everywhere.

$$L_{smooth} = \sum\nolimits_{(x,y)\in\mathbb{R}^{\mathbb{W}\times\mathbb{H}}} \rho\Big((\partial_x \widetilde{F^x_{12}}(x,y))^2 + (\partial_y \widetilde{F^x_{12}}(x,y))^2 + (\partial_x \widetilde{F^y_{12}}(x,y))^2 + (\partial_x \widetilde{F^y_{12}}(x,y))^2\Big) \tag{4.4}$$

where ∂_x and ∂_y are gradient along x and y direction, and $\widetilde{F^x_{12}}(x,y)$ and $\widetilde{F^y_{12}}(x,y)$ are the value of predicted flow field in x and y direction at point (x,y).
The total loss is a simple weighted sum of the brightness constancy loss (4.1), the gradient constancy loss (4.2), the smoothness loss (4.4), and the EPE loss (4.3),

$$\mathbf{L_{final}} = \lambda_1\mathbf{L_b} + \lambda_2\mathbf{L_g} + \lambda_3\mathbf{L_{epe}} + \lambda_4\mathbf{L_{smooth}} \tag{4.5}$$

where $\lambda_1, \lambda_2, \lambda_3$ and λ_4 tells the relative importance of each loss during training.

4.3 Training Details

Our network is trained end-to-end using Adam optimizer because for our task it shows faster convergence than standard stochastic gradient descent with momentum. The training converges after roughly a day. At input layer we stack two consecutive image to get input of size $\mathbb{R}^{H\times W\times 6}$. At loss layer, ground truth optical flow of size $\mathbb{R}^{H\times W\times 2}$ is used for loss calculation. As training loss we used L_{final} in equation (4.5), which is the weighted sum of the brightness constancy loss, the gradient constancy loss, the smoothness loss, and the standard EPE loss. and their correspnding weights are $(\lambda_1, \lambda_2, \lambda_3,\ \lambda_4)$. During training, we first assign equal weights to loss from different image scales and then progressively increase the weight on the larger scale image in a way similar to [26].Here we will use higher weights of image gradient photometric loss and second-order smoothness loss for KITTI because the data has more lightning changes and its optical flow has more continuously varying intrinsic structure. We also plot plot EPE loss during training to make decision on parameter tuning. We preformed iterative training 600k times with a batch size of 4 image pairs to train our network. We start with learning rate λ = 1e-4 and then divide it by 2 every 100k iterations after the first 300k. To monitor overfitting, we train our network with the mix of training data from MPI-Sintel and KITTI dataset and then fine-tune using Middlebury dataset.

Chapter 5

Experiments

5.1 Hardware and Software used

5.1.1 Hardware

Specifications:

- Linux Operating System version 4.15.0
- Intel® Xeon® Octa-Core E5-2667 CPU @ 3.20 GHz
- Nvidia Titan X Pascal 12 GB GDDR5X Graphic Card
- 128GB RAM 1600MHz DDR4
- Ubuntu 16.04.04 LTS (Linux Distribution)
- 256GB PCIe-based flash storage (configurable to 512GB flash storage)

5.1.2 Software

- Language used : Python 2.7.12
- Keras 2.3.1 with tensorflow 2.0.0 : Keras is a neural network library providing providing high-level APIs while TensorFlow is the used to provide low-level APIs for a number of various tasks in machine learning
- CV2 3.4.2 : OpenCV-Python is a library of Python bindings designed to solve computer vision problems. Used for processing image data.
- CUDA and CuDNN
- Matplotlib for visualising images and flow.

5.2 Parameter Tuning

Optimization using Adam optimizer uses four parameters.

- α, also referred to as the learning rate or step size. We start with learning rate α = 1e-4 and then divide it by 2 every 100k iterations after the first 300k.
- $\beta 1$, exponential decay rate for the first moment estimates.We fix the parameters as recommended in [9] = 0.9.
- $\beta 2$, exponential decay rate for the second-moment estimates. We fix the parameters as recommended in [9] - 0.999.
- ϵ, a very small number to prevent any division by zero

Other hyperparameter are for our loss function, L_{final} in equation (4.5) and have different value for different datasets used.

- λ_1, is the weight assigned to brightness constancy.
- λ_2, is the weight assigned to gradient constancy.
- λ_3, is the weight assigned to MSE error of predicted flow.
- λ_4, is the weight assigned to Flow Smoothness Loss.

These are set experimentally by trial and error method and vary from data-to-data. A few samples of parameter tuning along with the loss can be seen in Table 5.1.

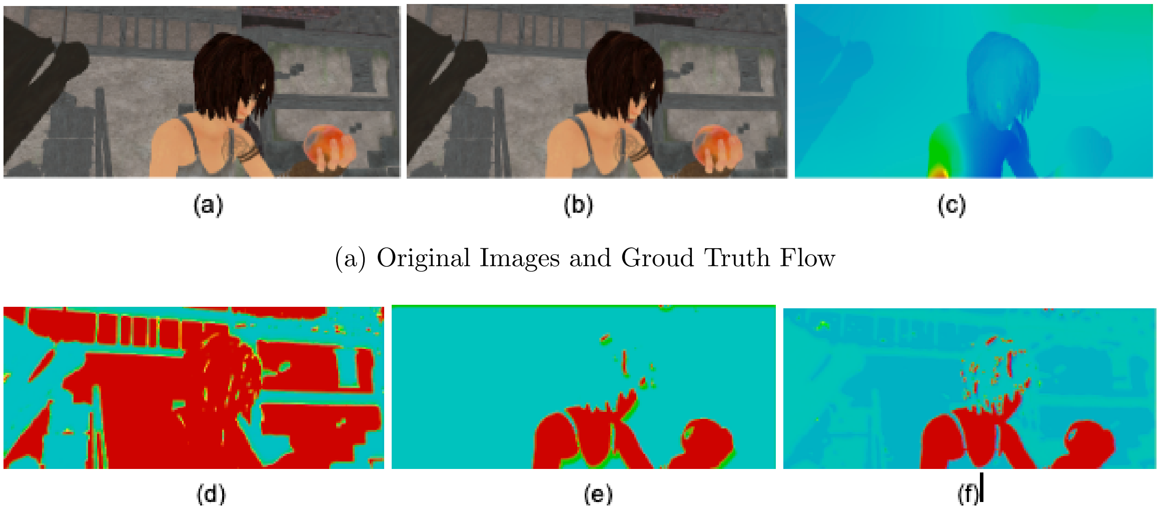
5.3 Experimental Results

We did partial training for parameter tuning with batch size, bs = 4 and 100 epochs. The results generated are just with MPI-Sintel Dataset with no data augmentation. Results of our parameter tuning ($\lambda_1, \lambda_2, \lambda_3,\ \lambda_4$) can be seen in Table 5.1.We used standard loss L_epe to describe our result.

As can be seen from table best parameter by trial and error method is (1, 0.1, 0.1, 1). Our gradient and smoothness weights are low and photometric and epe loss weights are high for MPI-Sintel Albedo data. This is because Albedo data flow in already smooth with very small variation in flow vectors and have brightness is constant almost everywhere except at edges. In Figure 5.1(f), we can see more details in flow variation are bein predicted properly when compared to Figure 5.1(d) and Figure 5.1(e).

Table 5.1: Parameter Tuning Table on MPI Sintel data containing weights for our four loss function , as in equation (4.5) and corresponding average L_{epe} loss reported while training.

λ_1	λ_1	λ_3	λ_4	Loss(L_{epe})
1	1	1	1	56.8
1	0.1	1	0.1	54.5
0.1	0.1	0.1	1	53.3
1	0.1	0.1	1	51.15



(a) Original Images and Groud Truth Flow

(b) Predicted Flow with Different Parameter Tuning

Figure 5.1: (a),(b) and (c) is Image1, Image2 and Ground Truth Flow. (d), (e) and (f) is predicted flow with our best parameter being for (f). (g) and (f) is another example of ground truth and predicted flow from our best parameter settings.

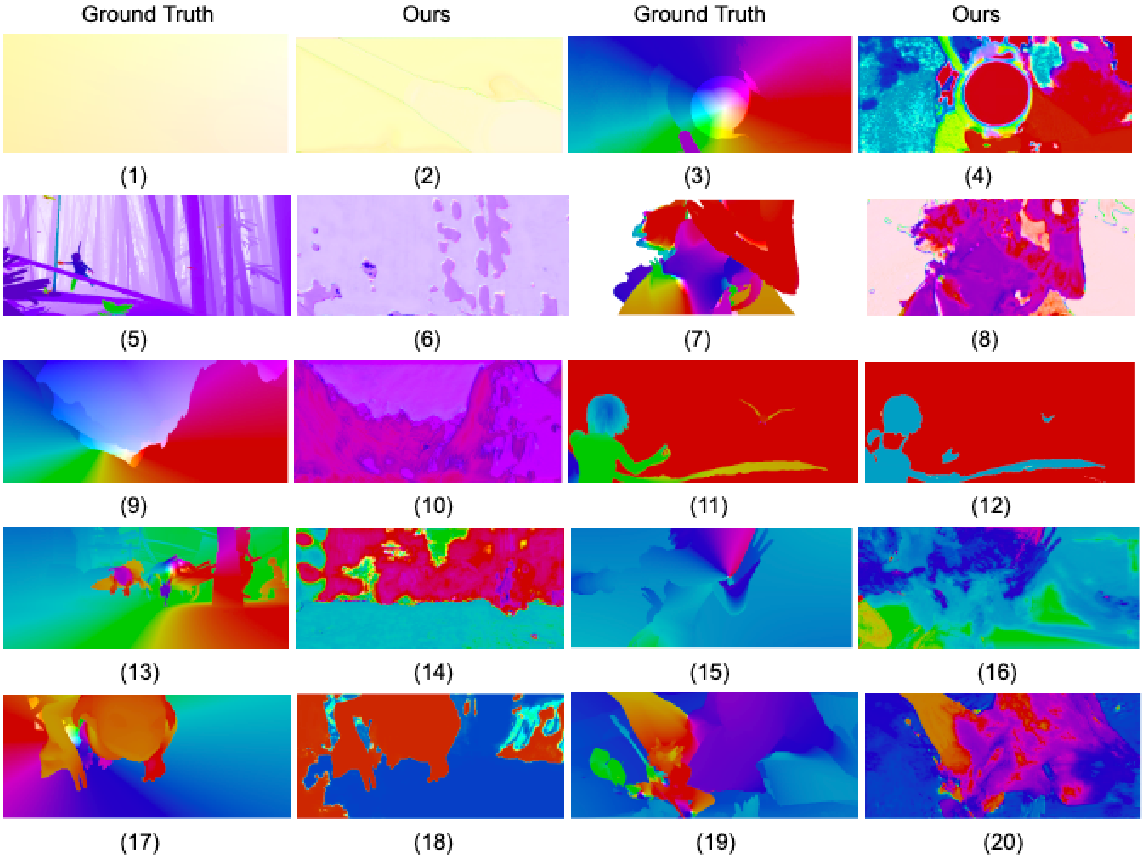


Figure 5.2: Displays Qualitative Results using random 10 image pairs from MPI-Sintel Dataset.

Table 5.2: Shows Quantitaive Result using MSE Error for 10 image pairs from MPI-Sintel Dataset.

Ground Truth Image) - (Predicted(Ours))	Root Mean Square Error(RMSE)
(1) - (2)	0.0928
(3) - (4)	1.1391
(5) - (6)	0.2217
(7) - (8)	0.4123
(9) - (10)	1.7975
(11) - (12)	0.3985
(13) - (14)	3.1904
(15) - (16)	2.8214
(17) - (18)	0.7839
(19) - (20)	0.4362

Further, in Figure 5.2, we have randomly sampled image pairs from Albedo version of MPI-Sintel dataset for our optical flow estimation and calculated Mean Sqauare Error in Table 5.2. Qualitatively, it can be seen our model is able to predict smooth flow regions quite well ,meaning no abrupt changes in motion between two images, as visible in Figure 5.2(1)-(2) where RMSE loss is very low. But our system struggles when their is abrupt changes, due to many different motion in between frames, as in Figure 5.2(13)-(14), where RMSE loss is very high. During training, our network only predicts forward flow, the total computational time for each epoch with 180 steps on MPI-Sintel Albedo 4 image pairs is roughly 84 seconds on Nvidia Titan Xp GPU and prediction time is around 100 milliseconds.

5.4 Comparison and Discussion

(TO BE UPDATED LATER)

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this thesis, we have used a very compact CNN model, U-Net, and leveraged from well-proven energy-based loss function namely, brightness, gradient, smoothness and EPE, as well as occlusion reasoning enabled by ground truth flow. Combining deep learning with domain knowledge not only reduces the model size but also improves the performance. The experimental results show that it can obtain clearer flow fields and can improve the accuracy of optical flow estimation. Going forward, our results suggest that further research on more accurate losses for supervised deep learning may be a promising direction for advancing the state-of-the-art in optical flow estimation. And some suggested future work, in next section, may help it improve even more.

6.2 Future Work

- We have done all prediction at coarse level, instead use Spatial-Pyramidal structure by stacking several Unet and predicting at all coarse-to-fine level, will help to extract large motion using coarse-to-fine approach with warping at each pyramidal level.
- We have used brightness value for loss computation, instead try using feature as features are less susceptible to shadows and lighting changes.
- We have predicted forward flow, try predicting backward flow by stacking image in reverse order and hence use it to produce better occlusion mask and optical flow using forward-backward consistency and occlusion-disocclusion symmetry.
- We have used consecutive image pairs , instead use multiple temporally consecutive image and Recurrent Unet Network.

Bibliography

[1] A.Bruhn, J.Weickert, C.Schnorr: Lucas/kanade meets horn/schunck: combining local and global optic flow methods. International Journal of Computer Vision (IJCV) (2005)

[2] A.Dosovitskiy, P.Fischery, E.Ilg, P.Hausser, C.Hazirbas, V.Golkov, P.D., T.Brox: Flownet: Learning optical flow with convolutional networks. in Proceedings of the 2015 IEEE International Conference on Computer Vision (2015)

[3] A.Geiger, P.Lenz, R.Urtasun: Are we ready for autonomous driving? the kitti vision benchmark suite. In CVPR (2012)

[4] A.Ranjan, M.J.Black: Optical flow estimation using a spatial pyramid network. in Proc.IEEE Conf.Comput.Vis.Pattern Recognit.(CVPR) (2017)

[5] B.Horn, B.Schunck: Determining optical flow.artificial intelligence. Artificial intelligence Vol.17, 1981–203 (1981)

[6] Butler, D.J., Wulff, J., Stanley, G.B., Black, M.J.: A naturalistic open source movie for optical flow evaluation. European Conf. on Computer Vision (ECCV) pp. 611–625 (Oct 2012)

[7] C.Bailer, B.Taetz, D.Stricker: Flowfields : Dense correspondence fields for highly accurate large displacement optical flow estimation. Proc.IEEE Int.Conf.Comput.Vis.(ICCV) (2015)

[8] C.Bailer, K.Varanasi, D.Stricker: Cnn-based patch matching for optical flow with thresholded hinge embedding loss. In CVPR (2017)

[9] D.P.Kingma, J.Ba.Adam: Adam: A method for stochastic optimization. In ICLR (2015)

[10] D.Sun, S.Roth, M.J.Black: A quantitative analysis of current practices in optical flow estimation and the principles behind them. International Journal of Computer Vision (IJCV) (2014)

[11] E.Ilg, N.Mayer, T.Saikia, M.Keuper, A.Dosovitskiy, T.Brox: Flownet 2.0: Evolution of optical flow estimation with deep networks. in Proc.IEEE Conf.Comput.Vis.Pattern Recognit (2017)

[12] F.Guney, A.Geiger: Deep discrete flow. In Asian Conference on Computer Vision Springer, 207—-224 (2016)

[13] I.Cohen: Nonlinear variational method for optical flow computation. In Proc.Eighth Scandinavian Conference on Image Analysis Vol.1, 523–530 (1993)

[14] I.Kajo, A.S.Malik, N.Kamel: An evaluation of optical flow algorithms for crowd analytics in surveillance system. Proceedings of the 2016 International Conference on Intelligent and Advanced Systems (2017)

[15] J.Hur, S.Roth: Mirrorflow: Exploiting symmetries in joint optical flow and occlusion estimation. In ICCV (2017)

[16] J.Janai, F.Guney, A.Behl, A.Geiger: Computer vision for autonomous vehicles: Problems, datasets and state-of-the-art. arXiv preprint arXiv:1704.05519 (2017)

[17] J.Xu, R.Ranftl, V.Koltun: Accurate optical flow via direct cost volume processing. In CVPR (2017)

[18] K.Simonyan, A.Zisserman: Two-stream convolutional networks for action recognition in videos. In Advances in Neural Information Processing Systems (NIPS (2014)

[19] Loncaric1, S., Majcenic1, Z.: Optical flow algorithm for cardiac motion estimation. IEEE (2002)

[20] L.Xu, J.Jia, Y.Matsushita: Motion detail preserving optical flow estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) (2012)

[21] M.Bai, W.Luo, K.Kundu, R.Urtasunr: Exploiting semantic information and deep matching for optical flow. In European Conference on Computer Vision Springer, 154–170 (2016)

[22] M.J.Black, P.Anandan: The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. Computer Vision and Image Understanding (CVIU) (1996)

[23] M.Menze, A.Geiger: Object scene flow for autonomous vehicles. In CVPR (2015)

[24] M.Neoral, J., J.Matas: Continual occlusion and optical flow estimation. In n Asian Conference on Computer Vision,Springer p. 159–174 (2018)

[25] N.Bonneel, J.Tompkin, K.Sunkavalli, D.Sun, S.Paris, H.Pfister: Blind video temporal consistency. ACM SIG- GRAPH pp. 34–196 (2015)

[26] N.Mayer, E.Ilg, P.Hausser, P.Fischer, D.Cremers, A.Dosovitskiy, T.Brox.: A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition pp. 4040—-4048 (2016)

[27] N.Papenberg, A.Bruhn, T.Brox, S.Didas, J.Weickert: Highly accurate optic flow computation with theoretically justified warping. International Journal of Computer Vision Vol.67, 141–158 (2006)

[28] O.Ronneberger, P.Fischer, T.Brox: U-net: Convolutional networks for biomedical image segmentation. in Proceedings of the 2015 International Conference on Medical Image Computing and Computer-Assisted Intervention (2015)

[29] P.Hu, G.Wang, Y.P.Tan: Recurrent spatial pyramid cnn for optical flow estimation. IEEE Trans.Multimedia Vol.20(10), 2814–2823 (2018)

[30] P.Weinzaepfel, J.Revaud, Z.Harchaoui, C.Schmid: Deep-flow: Large displacement optical flow with deep matching. IEEE International Conference on Computer Vision (ICCV) (2013)

[31] Q.Chen, V.Koltun: Full flow: Optical flow estimation by global optimization over regular grids. In CVPR (2016)

[32] Shi, Y.Q., Sun, H.: Image and video compression for multimedia engineering. CRC Press Vol. 1 (1999)

[33] S.Meister, J., S.Roth: Unflow: Unsupervised learning of optical flow with a bidirectional census loss. In Proc. AAAI Conf. Artif. Intell. (AAAI),New Orleans, LA, USA (2018)

[34] S.Zhao, Y.Sheng, Y.E., Y.Xu: Maskflownet: Asymmetric feature matching with learnable occlusion mask. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020)

[35] T.Brox, A.Bruhn, N.Papenberg, J.Weickert: High accuracy optical flow estimation based on a theory for warping. In T.Pajdla and J.Matas, editors, Proc.8th European Conference on Computer Vision Vol.3024 of LNCS, 25–36 (2004)

[36] T.Roosendaal(Producer): Sintel. blender foundation, durian open movie project (2010). `http://www.sintel.org/` (September 2010)

[37] T.W.Hui, X.Tang, C.C.Loy: Liteflownet: A lightweight convo- lutional neural network for optical flow estimation. in Proc.IEEE Conf.Comput.Vis.Pattern Recognit (2018)

[38] V.Vaquero, G.Ros, F.Moreno-Noguer, A.M.Lopez, A.Sanfeliu: Joint coarse-and-fine reasoning for deep optical flow. in Proc.IEEE Int.Conf.Image Process.(ICIP) (2017)

[39] Website, S.: `http://sintel.is.tue.mpg.de`

[40] X.Xiang, M.Zhai, R.Zhang, Y.Qiao, Saddik, A.: Deep optical flow supervised learning with prior assumptions. IEEE Access Vol.6 (2018)

[41] Y.Wang, Y.Yang, Z.Yang, et al., L.: Occlusion aware unsupervised learning of optical flow. arXiv preprint arXiv:1711.05890v2 (2018)

[42] Y.Zhu, S.Newsam: Densenet for dense flow. In Proc. IEEE Int. Conf. Image Process. (ICIP) p. 790–794 (2017)

[43] Z.Ren, J.Yan, B.B.X.a.H.: Unsupervised deep learning for optical flow estimation,. In Proc. AAAI Conf. Artif. Intell. (AAAI) pp. 1—-7 (2017)