Computer Vision | Homework 4
Review of Research Paper on Optical Flow Computation

Ruchi Manikrao Dhore - W1652116

Monday 13th February, 2023

1 Introduction

As part of this homework, we would study and review the paper *"FlowNet: Learning Optical Flow with Convolutional Networks"*. The authors of the research paper are *Philipp Fischer, Alexey Dosovitskiy, Eddy Ilg, Philip Häusser, Caner Hazırbaş, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, Thomas Brox.* Published by *arXiv* in the year 2015. We will provide an overview of the research area, the main research question, and the significance of the research. The key findings and contributions of the research papers will be reviewed.

2 Rationale For Research

Convolutional neural networks (CNNs) have become a successful method for image classification in a variety of computer vision tasks mostly related to recognition but not much success for **per-pixel based** optical flow estimation.

3 Major Objective Of Research

As a supervised learning task, the optical flow estimation problem can be solved using CNN. To learn how to anticipate the optical flow field from a pair of images, CNNs are trained end to end.

4 Previous Related Work

4.1 Few Papers from Previous Related Work

In 1981, Berthold K.P. Horn, Brian G. Schunck presented their work on determining optical flow by considering two measurements namely space and time. In 1989, Y. LeCun et. al applied Backpropagation to handwritten zip code recognition. In 1998, E. M´emin et. al presented a model to couple the motion estimation process with an object-based motion segmentation. In 2004, T. Brox et. al applied theory of warping to obtain high accuracy of optical flow estimation. In 2012, A. Krizhevsky et. al used Deep CNN for image classification. In 2012, D. Eigen et. al used a multi-scale deep network for depth estimation of single image. In 2015, J. Long presented the use of fully CNN for semantic segmentation.

4.2 Summary of Previous Related Work using Machine Learning

Several authors have applied machine learning techniques such as Gaussian Mixture, principal component analysis to optical flow before. There has been research on employing neural network models for unsupervised learning of disparity or motion between video frame.

4.3 Summary of Previous Related Work using Deep Learning

If there is a sufficient amount of labeled data, CNNs are known to be particularly good at learning input-output relations. Convolutional neural networks are used with backpropagation and perform well on large-scale image classification by Krizhevsky et al. This is the beginning to apply CNN to various computer vision tasks. From CNNs trained in a supervised or unsupervised manner, Fischer et al. extract feature representations and match these features using Euclidean distance. Zbontar et. al trained a CNN with a Siamese architecture to predict similarity of image patches. Recent applications of CNNs include semantic segmentation, prediction , keypoint prediction and edge detection. These challenges include per-pixel predictions, which makes them comparable to optical flow estimates. In order to compute a single prediction for each input image patch, a traditional CNN is used in a "sliding window" approach, which has the advantage of being simple but has disadvantages like as expensive computational. The value of interest can be predicted using a concatenated per-pixel feature vector, which is created by upsampling all feature maps to the necessary full resolution and stacking them together. By training a second network with the coarse prediction and the input image as inputs, Eigen et al. improved a coarse depth map.

5 Author Approach

This approach integrated ideas from both Long et al and Dosovitskiy et al. and presented two architectures. In first architecture, they used **'upconvolve' the whole coarse feature maps**, allowing to transfer more high-level information to the fine prediction. In the second architecture, they concatenated the **'upconvolution' results with the features from the 'contractive' part** of the network.

5.1 Network Architecture Details

In order to forecast optical flow over a dataset made up of image pairings and ground truth flows, the author proposed using a textbf end-to-end learning technique. Directly from the photos, they trained a network to predict the x-y flow fields. In order to make network training computationally practical, which enables the aggregation of information over significant portions of the input images, pooling in CNNs is known to be required. However, pooling reduces resolution. Therefore, in order to provide dense per-pixel predictions, there is a need to refine the coarse pooled representation. This is done in two parts: **First is contraction and latter is expansion**. Backpropagation is used to train networks as a whole that have sections that are contracting and increasing.

5.2 Architecture 1: FlowNetSimple

Generic architecture consists only of convolutional layers. In order to analyse the image pair and extract the motion information, they stacked the two input images together and fed them through a generic network. Figure 1 depicts FlowNetSimple architecture.

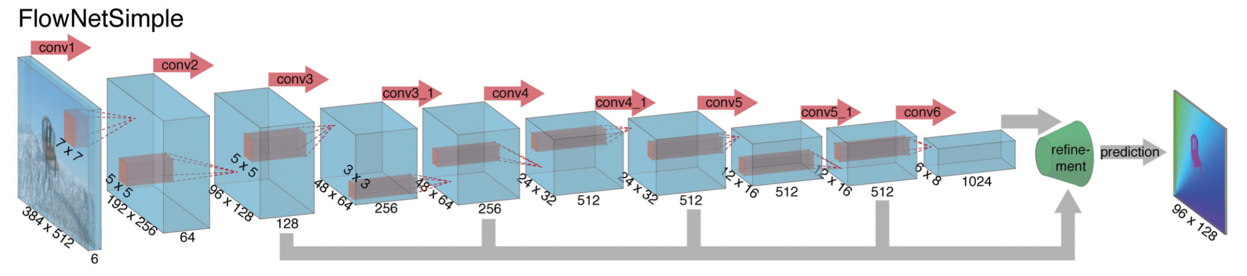

Figure 1: FlowNetSimple Architecture

5.3 Architecture 2: FlowNetCorr [Contracting and Expanding Parts]

In this method, the two images were split into two distinct, identical processing streams, which were then combined. This architecture first produces meaningful representations of the two images separately and then combined them on a higher level. This is analogous to the standard matching approach where it first extracts features from patches of both images and then compares those feature vectors. Figure 2 depicts FlowNetCorr architecture.

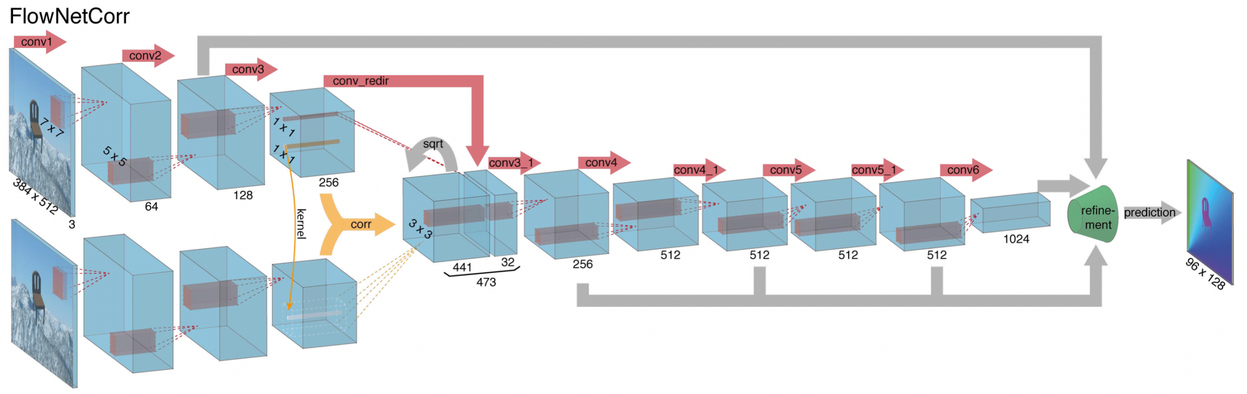

Figure 2: FlowNetCorr Architecture

5.3.1 Contracting Part

In Contracting part, they added a "correlation layer" that carries out multiplicative patch comparisons between two feature maps for the matching process. A correlation layer enables the network to compare each patch from f1 with each path from f2, given two multi-channel feature maps f1, f2: R2 Rc, with w, h, and c being their width, height, and number of channels. They only take into account one comparison between two patches. The following equation for a square patch of size K:= 2k+1 defines the "correlation" of two patches that are centered at x1 in the first map and x2 in the second map.

$$c(\mathbf{x}_1, \mathbf{x}_2) = \sum_{\mathbf{o} \in [-k,k] \times [-k,k]} \langle \mathbf{f}_1(\mathbf{x}_1 + \mathbf{o}), \mathbf{f}_2(\mathbf{x}_2 + \mathbf{o}) \rangle$$

In this equation, data is convolved with other data rather than using a filter, as in a convolution step in neural networks. You must multiply by c*K2 in order to calculate c(x1, x2). Such computations are required for comparing all patch combinations, which results in a sizable result and renders efficient forward and backward passes impractical. Therefore, they restrict the greatest displacement for comparisons for computational reasons. They restricted the range of x2, and for each location x1 they computed correlations c(x1, x2) only in a neighborhood of size D:= 2d + 1. Strides s1 and s2 were employed to quantize x1 globally and x2 specifically within the area surrounding x1.

5.3.2 Expanding Part

The "upconvolutional" layers, which are made up of an unpooling and a convolution, are the main component of the expanding part. In order to enhance the feature maps, they utilized the "upconvolution" technique, joined the feature maps together with those from the "contractive" portion of the network and an up-sampled coarser flow forecast. They were able to maintain both the fine local information offered by lower layer feature maps and the high-level information transmitted by coarser feature maps in this fashion. The resolution was enhanced twice with each step. They did this four times, producing a predicted flow whose resolution is still four times less than the input. Figure 3 depicts refinement part from Figure 2.

5.3.3 Variational Refinement

In this variation, instead of bilinear upsampling, they used the variational approach without the matching term. They employed the coarse to fine approach with 20 iterations using images that had been four times downscaled in quality to fully resolve the flow field. At the full image resolution, they then perform 5 additional rounds. Although this upscaling technique requires more computer power than straightforward bilinear upsampling, it combines the advantages of variational techniques to produce flow fields that are smooth and subpixel precise.

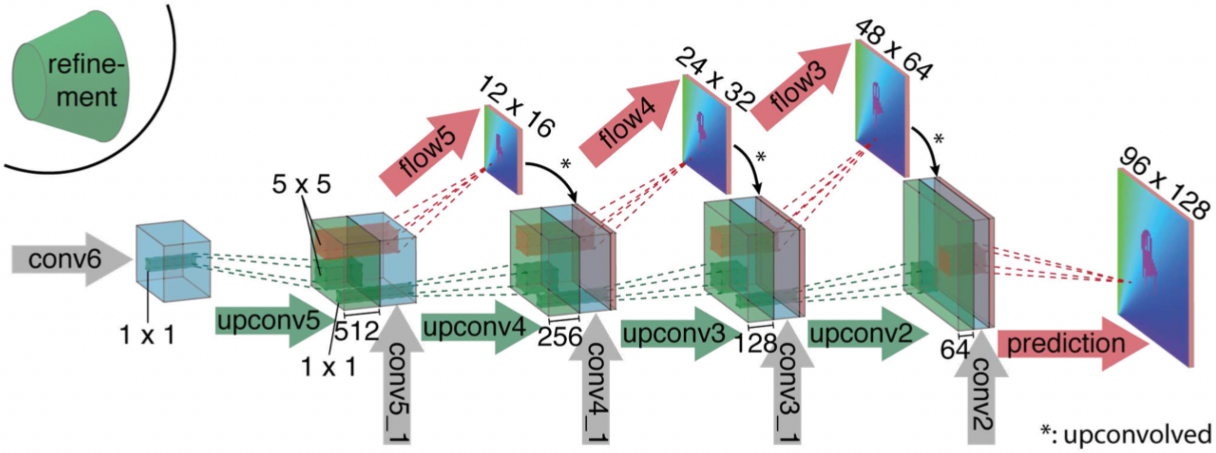

Figure 3: Refinement.png

6 Dataset Used

Several datasets were used. For generated dataset, they also performed data augmentation. Following table lists the same:

Category	Description
Available	Middlebury, KITTI , Sintel
Generated	Synthetic Flying Chairs of 22,872 frame pairs as well as frames with ground truth

Table 1: Dataset

7 Experimentation

For hardware support, training and testing of the networks was performed on NVIDIA GTX Titan GPU. As well carried out on the Sintel, KITTI and Middlebury datasets, as well as on Synthetic Flying Chairs dataset. Also experimented with fine-tuning of the networks on Sintel data and variational refinement of the predicted flow fields.

8 Results

Following are the results of the experimentation:

FlowNetC is better than FlowNetS on Sintel dataset
FlowNetS outperforms FlowNetC on KITTI dataset
FlowNetC outperforms FlowNetS on Flying Chair dataset

9 Research Outcome

They proposed and compared two architectures. The first one was a generic architecture and the second one was an architecture with a correlation layer that explicitly provides matching capabilities.

They demonstrated that networks trained on these fictitious datasets still generalize to them quite well, achieving competitive accuracy at frame speeds of 5 to 10 fps.