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

Use of CNN to solve the **per-pixel based** optical flow estimation problem as a supervised learning task. training CNNs end to end to learn predicting the optical flow field from a pair of images.

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 work on unsupervised learning of disparity or motion between frames of videos using neural network models.

4.3 Summary of Previous Related Work using Deep Learning

CNNs are known to be very good at learning input–output relations if there is enough labeled data. 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. Fischer et al. extract feature representations from CNNs trained in supervised or unsupervised manner and match these features based on 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 tasks are similar to optical flow estimation in that they involve per-pixel predictions. The simplest solution is to apply a conventional CNN in a 'sliding window' fashion, hence computing a single prediction for each input image patch but has drawbacks as high computational Another simple approach is to up-sample all feature maps to the desired full resolution and stack them together, resulting in a concatenated per-pixel feature vector that can be used to predict the value of interest. Eigen et al. refined a coarse depth map by training an additional network which gets as inputs the coarse prediction and the input image.

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 Architecture 1: FlowNetSimple

Generic architecture consists only of convolutional layers. In this they stacked both input images together and feed them through a generic network, to process the image pair to extract the motion information. Fig 1 depicts FlowNetSimple architecture.

6 Network Architecture Details

Author considered an **end-to-end learning approach to predict optical flow over a dataset consisting of image pairs and ground truth flows**. They trained a network to predict the x–y flow fields directly from the images. It is known that pooling in CNNs is necessary to make network training computationally feasible which allows aggregation of information over large areas of the input images. But pooling results in reduced 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**. Networks consisting of contracting and expanding parts are trained as a whole using backpropagation.

6.1 Architecture 1: FlowNetSimple

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

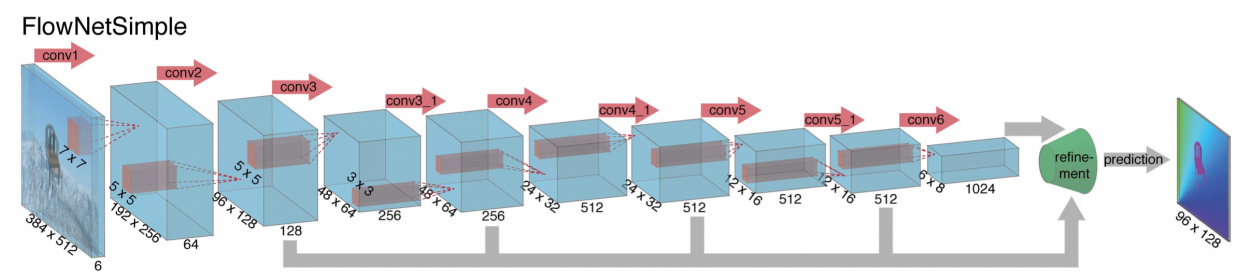

Figure 1: FlowNetSimple Architecture

6.2 Architecture 2: FlowNetCorr [Contracting and Expanding Parts]

In this approach they created two separate identical processing streams for the two images and then combined them at a later stage. 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.

6.2.1 Contracting Part

For the matching process, they introduced a 'correlation layer' that performs multiplicative patch comparisons between two feature maps. Given two multi-channel feature maps f1, f2 : R2 → Rc, with w, h, and c being their width, height and number of channels, a correlation layer lets the network compare each patch from f1 with each path from f2. They consider only a single comparison of two patches. The 'correlation' of two patches centered at x1 in

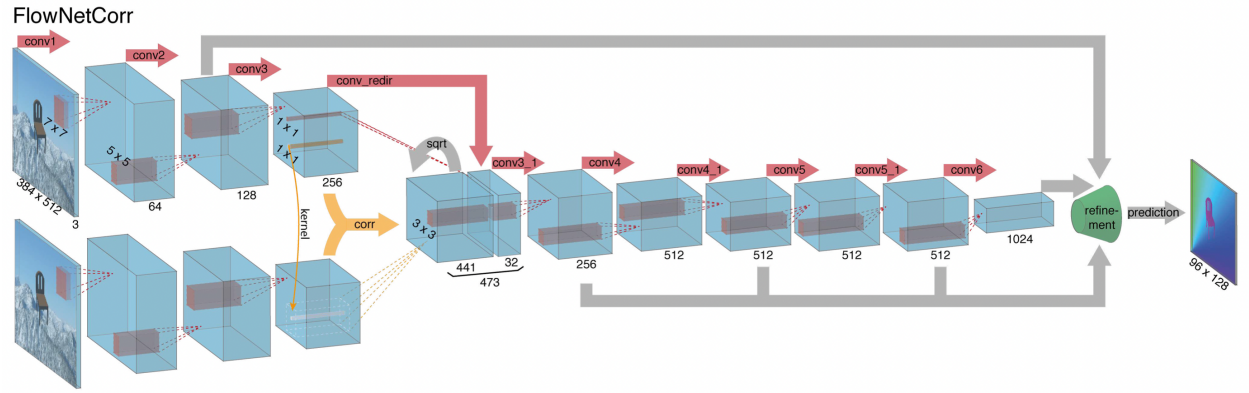

Figure 2: FlowNetCorr Architecture

the first map and x2 in the second map is then defined as the below equation for a square patch of size K:= 2k+1.

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

This equation is a step of a convolution in neural networks, but instead of convolving data with a filter, it convolves data with other data. Computing c(x1, x2) involves c*K2 multiplications. Comparing all patch combinations involves w2 h2 such computations, yields a large result and makes efficient forward and backward passes intractable. Thus, for computational reasons they limit the maximum displacement for comparisons . Given a maximum displacement d, for each location x1 they computed correlations c(x1, x2) only in a neighborhood of size D := 2d + 1, by limiting the range of x2. They used strides s1 and s2, to quantize x1 globally and to quantize x2 within the neighbourhood centered around x1.

6.2.2 Expanding Part

The main ingredient of the expanding part are 'upconvolutional' layers, consisting of unpooling and a convolution. To perform the refinement, they applied the 'upconvolution' to feature maps, and concatenate it with corresponding feature maps from the 'contractive' part of the network and an up-sampled coarser flow prediction. This way they preserved both the high-level information passed from coarser feature maps and fine local information provided in lower layer feature maps. Each step increased the resolution twice. They repeated this 4 times, resulting in a predicted flow for which the resolution is still 4 times smaller than the input. Figure 3 depicts refinement part from Figure 2.

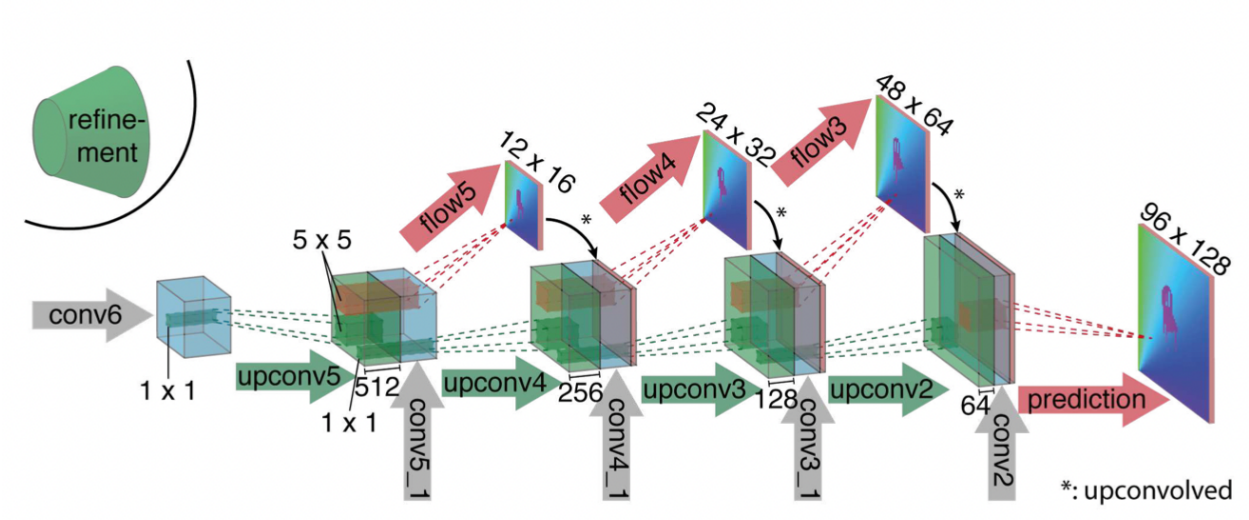

Figure 3: Refinement.png

6.2.3 Variational Refinement

In this variation, instead of bilinear upsampling, they used the variational approach without the matching term. They took 4 times downsampled resolution images and then used the coarse to fine scheme with 20 iterations to bring the flow field to the full resolution. Finally, they run 5 more iterations at the full image resolution. This upscaling method is more computationally expensive than simple bilinear upsampling, but adds the benefits of variational methods to obtain smooth and subpixel accurate flow fields.

7 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

8 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.

9 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

10 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 showed that networks trained on this unrealistic data still generalize very well to existing datasets such as Sintel and KITTI, achieving competitive accuracy at frame rates of 5 to 10 fps.