Chapter 19 – Video Motion Estimation

Author: Gianmarco Scarano

gianmarcoscarano@gmail.com

1. Video Motion Estimation

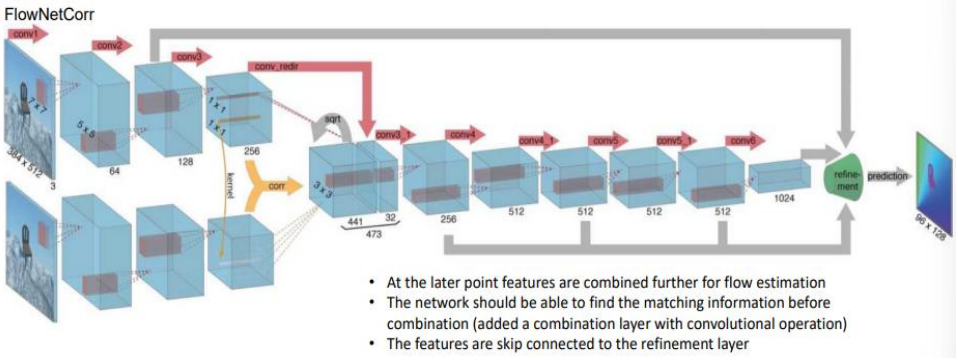
In Video Motion estimation, we compute the optical flows which are a distribution of apparent velocities of movement of brightness patterns in an image. If we compute the optical flow for subsequent frames, we can determine the displacement vector for each pixel.

When applying Optical Flow with Neural Networks, we have to remind that the classical approaches have some complex optimization problems, resulting in a computationally expensive task. That's why, when dealing with DNN, experts thought about having a trade-off between the size/complexity of the model and its accuracy.

1.1 FlowNet

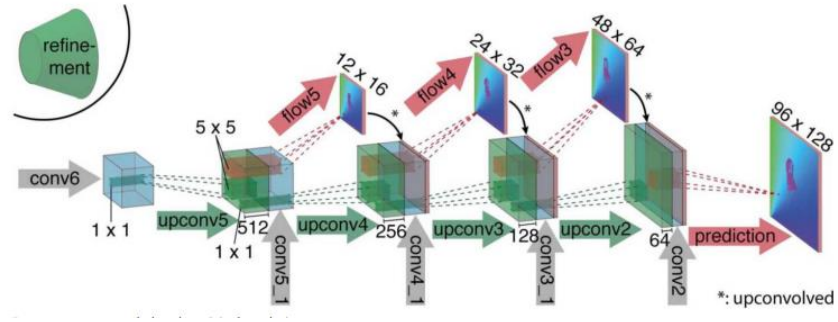
FlowNet is an end-to-end supervised learning for Optical Flow estimation. It's supported by an Encoder/Decoder architecture. The main concept behind this idea, relies on the fact that given two consecutive frames, the network has to find if an object in the first image is at another location in the second image. The two frames, though, need to match features at different location in input images. FlowNet also splits in **FlowNetSimple** (where we simply stack both input images together and let them go through a generic Convolutional step), or **FlowNetCorr**.
As for **FlowNetCorr**, we create two separates' streams for the two images and combine them at a certain stage.

FlowNetCorr (Encoder)



Really important is the yellow arrow, where we combine the two convolutions into a single one (**Correlation Layer**).

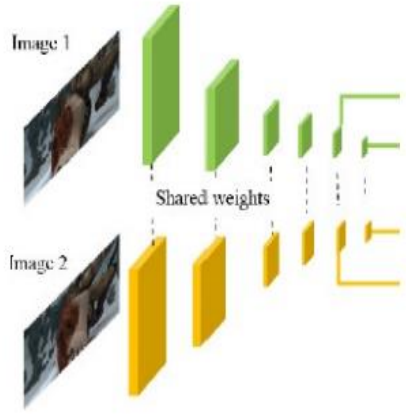
Upconvolutional Layers (Decoder)



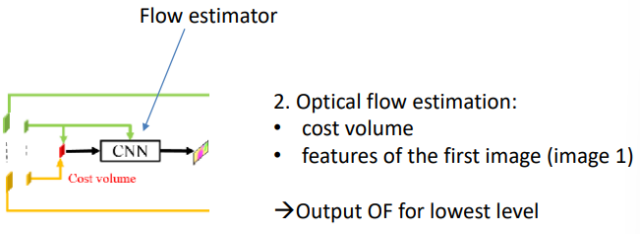
Here we have Unpooling features maps + convolution. What we do here, is to concatenate the Upconvolutioned Feature Maps with the corresponding map from the Encoder part.

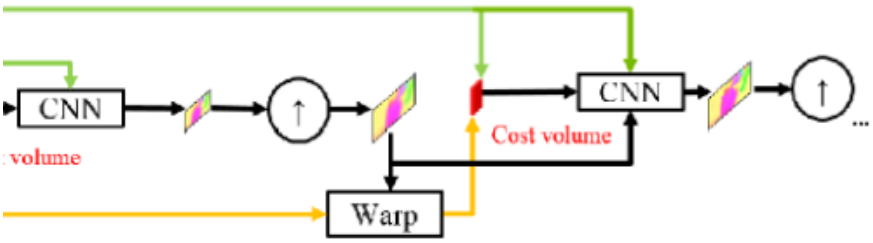
1.2 PWC-Net

PWC-Net is another architecture which computes optical flows. Here, we do feature extraction from input images with feature pyramid. Also, for every level of feature pyramid we compute the optical flow estimation. A few points for the PWC-Net:



- **Shared weights**: Find most similar pixels in features w.r.t other images. In this way learning the features pyramid at lower scale, allow us to learn global context in order to do the matching.





So, in the end, we compute the optical flow and we upsample it to match the initial spatial dimensions. We also warp the features of the 2nd image towards the features of the 1st image in order for the 2nd image to become more similar to the 1st. PWC-Net is actually, the state-of-the-art with end-to-end training. Also suitable for mobile applications.