Chapter 18 – Action Recognition

Author: Gianmarco Scarano

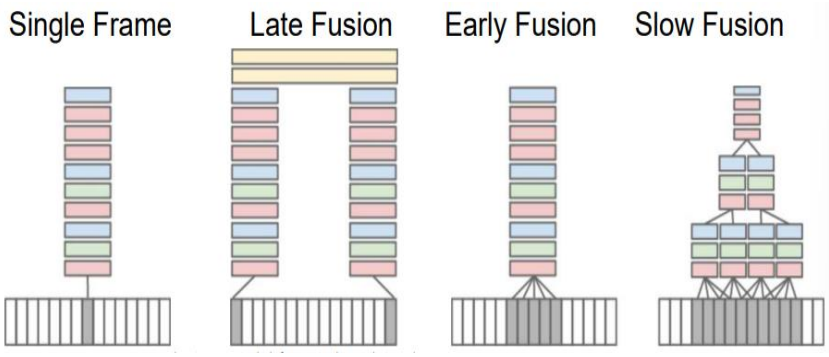
gianmarcoscarano@gmail.com

1. Video Classification

When talking about Video Classification we talk about the task of identifying a pre-defined set of physical actions (gestures, actions, interactions, group activities etc.).

We know that a video is a Tensor of shape [T, W, H, C] where T is the Temporal coordinate of a certain frame [W, H, C]. We could think, of course, to feed a CNN with each frame, but pooling operations is not aware of the temporal order of the frames.

So, multiple algorithms have been approached, with one involving fusing information over temporal dimension through the Network.



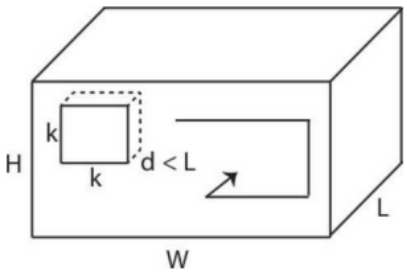
- **Late fusion**: Take one frame at time t, $t+1$, etc. and putting a CNN on top and some other layers which do the fusion work (the yellow ones)
- **Early fusion**: Here we take as input frames next to each other. In this way we are able to combine information.
- **Slow fusion**: Information of each frame is considered in one stream of the Network and is, then, combined with another stream and so on and so forth. After a block of Convolution, there is the fusion of the information coming from both streams (combination of infos).

Slow Fusion has been the best in terms of Video metrics, such as "**Video Hit@1**", "**Video Hit@5**" with 60.9 and 80.2 respectively.

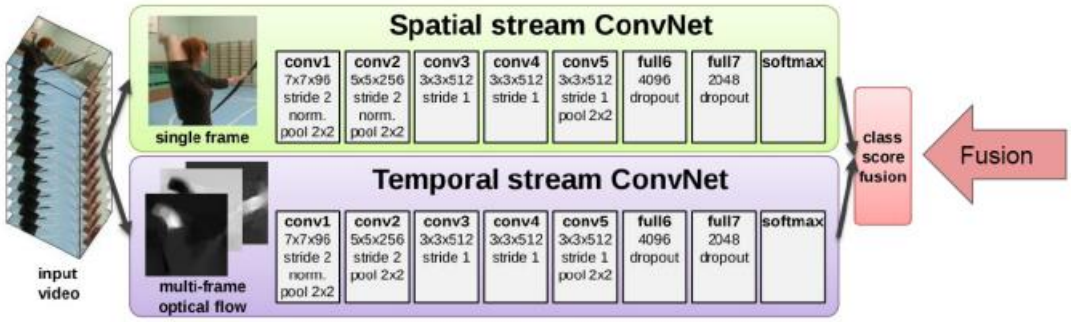
2. 2DCNN+RNN / C3D / etc.

As we know, in CNN we have the enormous limitation of predicting a sample at time t without being aware of previous samples at time $t-1$, $t-2$, etc. That's why RNN have been invented, since here hidden layers and output layers depend from the previous states of the hidden layers.

- **2DCNN + RNN**: Is a combination of a CNN and RNN (in RNN infos are coming from previous frames), such that we can process sequences of input, but since RNN are sequential we cannot parallelize them.
- **3D CNN (C3D)**: We add an extra dimension to the standard CNN. We don't have a Convolution for a single 2D frame, but we have a 3D Convolution sliding all over the frames (and so on the video).



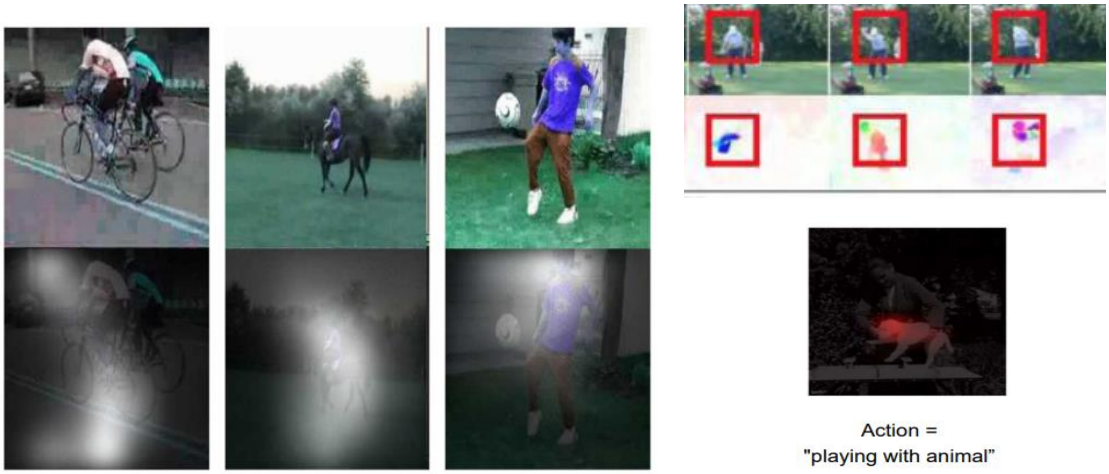
- **Two-streams 2D CNNs**: Here, we extract the optical flow for a stack of frames and use it as an input to a CNN. Along this, we extract single frames and we classify it normally as we always do with a CNN. At the end of the architecture, we join the two scores. We can also add a RNN before applying the softmax (*Two-streams 2D CNNs + RNN*).



- **Two-streams 3D CNNs:** It's the same as the Two-streams 2D CNN, but we switch to a 3D Convolutional Network.

3. Action Recognition

In Action Recognition tasks, we could deploy a R-CNN, along with a network with Attention, Soft-Attention or Hard-attention:

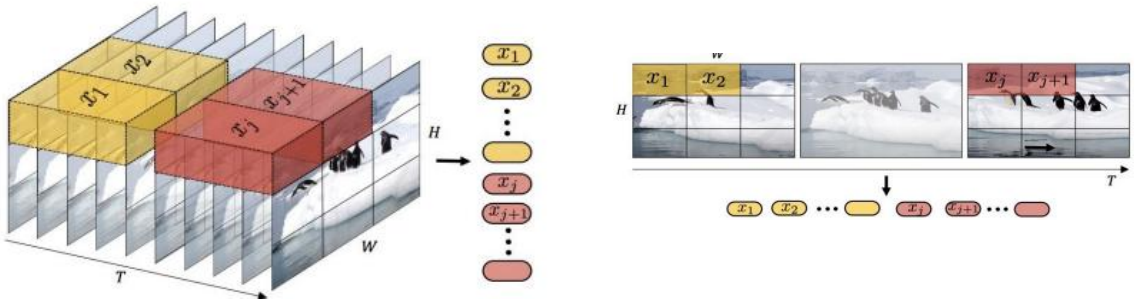


- Hard attention: The attention focus on a single input location, we can't use gradient descent and we need reinforcement learning
- Soft attention: We compute a weighted combination (attention) over inputs using an attention network.

3.1 Video ViT, Datasets & issues

In Video ViT, we apply a ViT to videos (well...). We sample n_t frames and we embed each 2D frame independently (tubelet architecture), following the ViT architecture that we are used to.

Here's the scheme:



Talking about Datasets, we remember the UCF-101 (101 categories from YouTube), Sports 1M from Stanford and Kinetics. As for issues, instead, currently GPUs can handle batches of 32/64 images (at once). We can also freeze some layers, in order to reduce the memory. We also must pay attention to GPUs, especially for the I/O bottleneck, where there will be a **OOM** (Out of Memory) issue, if we don't lower the batch size or we keep things as simple as possible. Finally, using asynchronous data loading pipelines is a really great technique for avoiding such issues.