

Video Classification with Convolutional Neural Network

A lite version of the Stanford White Paper

Riccardo Zuliani 875532

University Ca'Foscari of Venice
Department of Environmental Sciences, Informatics and Statistics

May 28, 2023



Ca' Foscari
University
of Venice

Goal:

Given a sequence of frames from a YouTube video predict the correct sport label

Today's Topics:

- Sports 1 Million Youtube Video Dataset
- Dataset & Problem Simplification
- Frames Download Strategy
- Two models types, with different input size
 - **Time Information Fusion Models**, takes *multiple frames*
 - **Multi-Resolution Models**, takes a *single frame*
- Results



- 1 million sport YouTube videos links annotated with 487 classes [2]
- 1000-3000 videos per class
- 7% (as of 2016) of the videos have been removed by the YouTube uploaders since the dataset was compiled
- 5% of the videos are annotated with more than one class
- Possible noise:
 - Data manually weakly annotated
 - Video may include nonsense frames
- **Multi-Label Classification Problem**



- Huge Dataset
- Choose **10 sports classes** →
- Consider all videos from the chosen classes
- **Multi-Class Classification Problem**

- rugby
- formula racing
- beach volleyball
- basketball
- karate
- motocross
- kitesurfing
- motorcycle racing
- horse racing
- bodybuilding

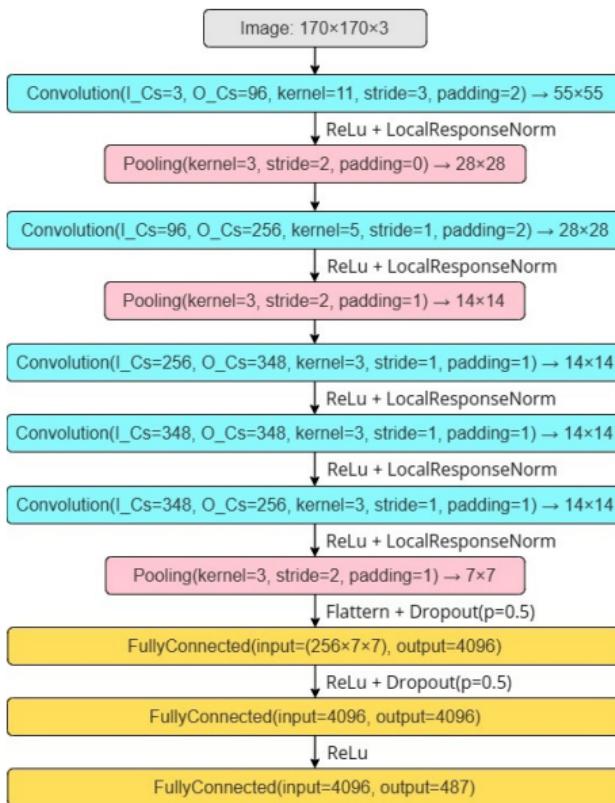
- **yt-dlp** a youtube-dl fork to obtain video info
- Download frames from videos with at least 20 FPS, that have **144p resolution** and the URL different from the manifest file
- Remove the **10%** of the *initial* and *final* video frames
- Sample the **1%** of contiguous and equidistant **half FPS** frames (aka bag of shots) for the remaining frames of each video
- **OpenCV** to download the contiguous bags of shots
- Zip and upload the dataset in **Google Drive**



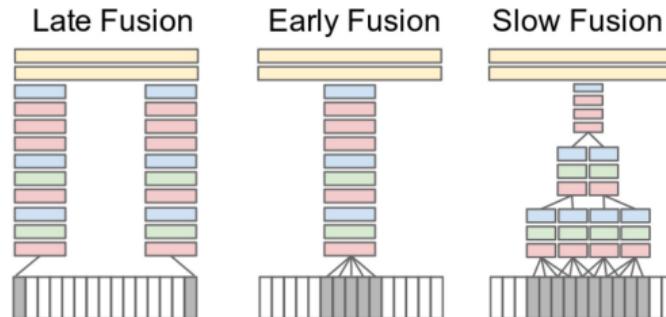
YT-DLP A youtube-dl fork with additional features and fixes



Primile CNN Architecture

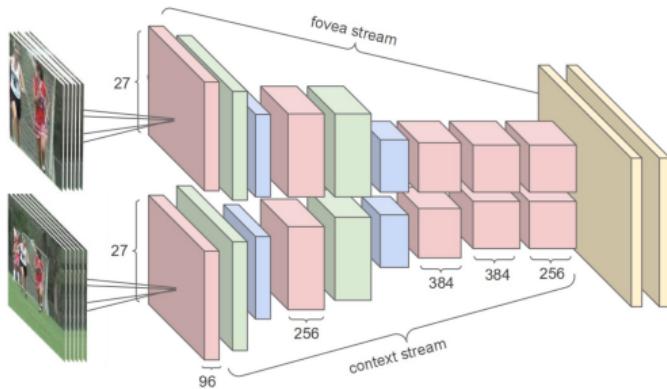


Time Information Fusion (aka Multi Frames) Models



- **Late Fusion:** two separate single-frame network with only the first FC layer be able to compute global motion characteristics
- **Early Fusion:** combines information across an entire time window immediately on the pixel level
- **Slow Fusion:** it slowly fuses temporal information throughout the network such that higher layers get access to progressively more global information in both spatial and temporal dimensions

Multi-Resolution Models (Fovea, Context, Fovea + Context)



- The **context stream** receives the downsampled frames at half the original spatial resolution (89×89 pixels)
- The **fovea stream** receives the center 89×89 region at the original resolution
- Since the input is half the spartial size, the last pooling layer is removed to ensure both streams terminating in a layer of $7 \times 7 \times 256$

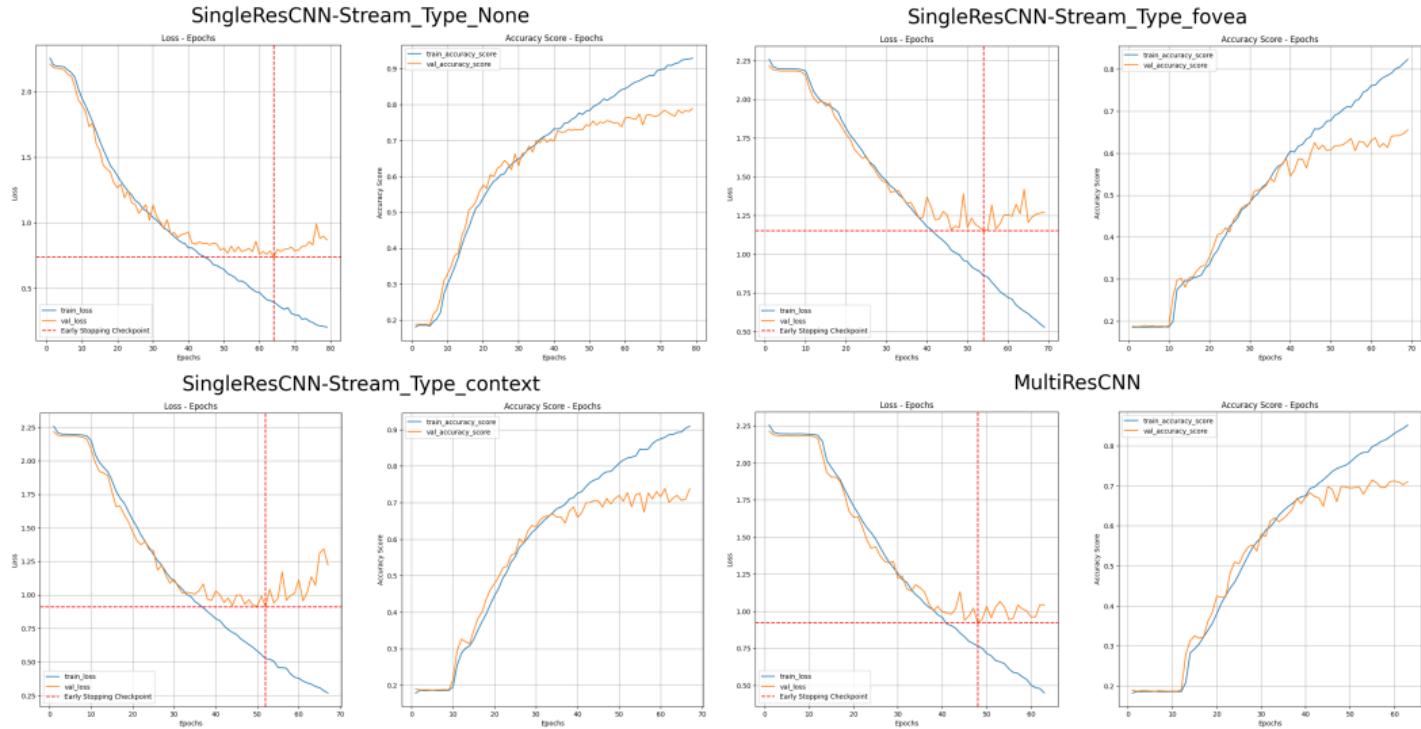
- 251038 training frames and 54247 testing frames
- 18287 training *bags of shots* and 3868 testing *bags of shots*
- **20%** of the train set assigned as **validations** set
- All models are trained up to **80 Epochs**
- **Stochastic Gradient Descent** as optimizer with *learning rate* $1e^{-3}$, *momentum term* 0.9 and *weight decay* $5e^{-5}$
- **ReduceLROnPlateau** as scheduler
- **Early stopping** with patience of 15 epochs

Also using a subset of 10 classes the results were subject to the **previous cited problems**.

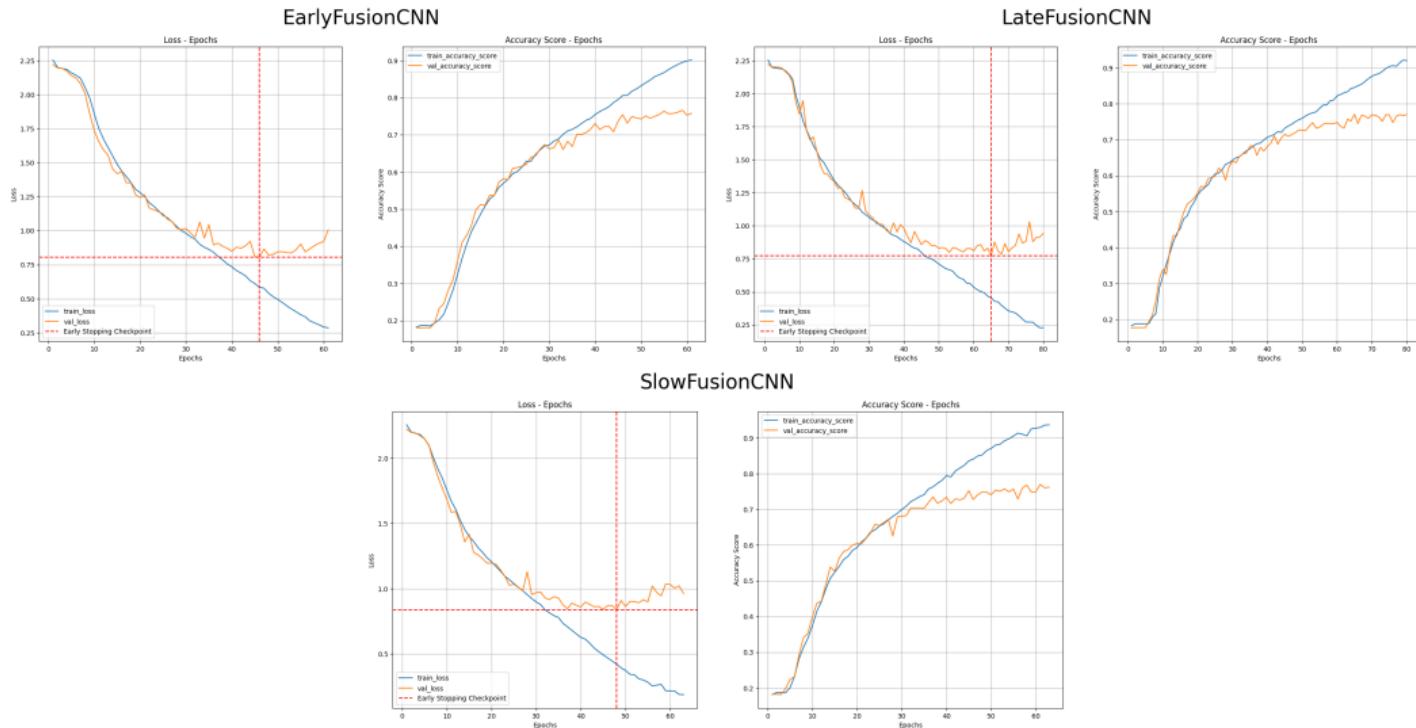
- *Wrong manually labelled video*
- *Useless sequence of frames due to pseudo-random bag of shots sampling*

Therefore the results are to be considered as **experiments** as dealing with these problems would require a lot of **manual work which would take too much time**.

Single Frame Results (Single Frame, Fovea Stream, Context Stream, MultiRes)



Multi Frame Results (Early Fusion, Late Fusion, Slow Fusion)



Summary

| Model Name | Test Loss | Test Acc | Test time min | Train time hr |
|--------------------|-----------|----------|---------------|---------------|
| <i>SingleRes</i> | 0.96767 | 0.71325 | 0.37903 | 1.53495 |
| <i>SingleRes_F</i> | 1.26127 | 0.58379 | 0.17914 | 1.20548 |
| <i>SingleRes_C</i> | 1.17831 | 0.65533 | 0.16217 | 1.16977 |
| <i>MultiRes</i> | 1.12028 | 0.63378 | 0.13316 | 1.52294 |
| <i>LateFusion</i> | 1.05671 | 0.69669 | 0.37975 | 3.01682 |
| <i>EarlyFusion</i> | 1.08308 | 0.67226 | 0.67731 | 3.79941 |
| <i>SlowFusion</i> | 1.27412 | 0.6623 | 1.42234 | 7.80681 |

Table 1: Obtained Results [3]

| Model Name | <i>SingleRes</i> | <i>SingleRes_F</i> | <i>SingleRes_C</i> | <i>MultiRes</i> | <i>LateFusion</i> | <i>EarlyFusion</i> | <i>SlowFusion</i> |
|------------|------------------|--------------------|--------------------|-----------------|-------------------|--------------------|-------------------|
| Test Acc | 0.411 | 0.3 | 0.381 | 0.424 | 0.407 | 0.389 | 0.419 |

Table 2: Stanford Results over the whole set of 1 million video [1]

- [1] Andrej Karpathy et al. “Large-scale Video Classification with Convolutional Neural Networks”. In: *CVPR*. 2014.
- [2] Andrej Karpathy et al. *The YouTube Sports-1M Dataset*. URL:
<https://github.com/gtoderici/sports-1m-dataset>.
- [3] Riccardo Zuliani. *VideoClassification-CNN*. URL:
<https://github.com/zuliani99/VideoClassification-CNN>.