Action spotting on SoccerNet Challenge

Advanced techniques

Presentation

MSc Artificial Intelligence for Science and Technology

Advanced techniques

Submitted by **Daniele Cecca Mat.914358**

UNIVERSITÀ DEGLI STUDI DI MILANO BICOCCA

Introduction

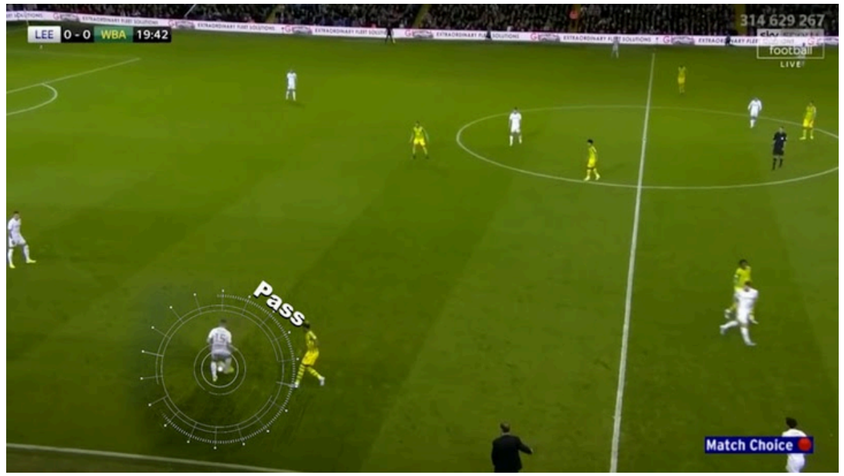
This project focuses on the development and implementation of **classification system for ball events in a football match**, also called ball **action spotting**.

It utilizes the **SoccerNet dataset** and draws inspiration from previously proposed solutions

PROJECT STRUCTURE

1. Load create and explore the **dataset**
2. Create the **ActionRecognitionDataset**
3. Design and develop the **architecture of the network.**
4. Train the network using different **hyperparameters**.
5. Select the **best-performing network** and train it for additional epochs.
6. **Evaluate** the network's performance.
7. Implementation of a web app with **Streamlit**

SoccerNet Dataset



The dataset is composed of **7 videos of English Football League games**, and to each video a **JSON** with the **timestamp** and the **label** is associated.

In total we have have **12 different types of action**

Pass	**Drive**	**Header**	**High Pass**
Out	**Cross**	**Throw in**	**Shot**
Ball Player Block	**Player Successful Tackle**	**Free Kick**	**Goal**

SoccerNet Dataset

CREATION OF THE DATASET

01

Divide the videos matches into segments

02

Create the dataframe

I will apply label encoding on the label

Since defining precise **temporal boundaries** for actions is challenging because it's hard to fix the exact **start** and **end** times, and knowing what occurs after the action can be beneficial, I **extend** the interval by one **second beyond** the defined action times in the JSON file.

	clip_filename	label	clip_duration
0	/content/data/new_val_data/2019-10-01 - Middle...	PASS	1
1	/content/data/new_val_data/2019-10-01 - Middle...	DRIVE	2
2	/content/data/new_val_data/2019-10-01 - Middle...	PASS	3
3	/content/data/new_val_data/2019-10-01 - Middle...	DRIVE	4
4	/content/data/new_val_data/2019-10-01 - Middle...	HIGH PASS	7

Submitted by Daniele Cecca

SoccerNet Dataset

EXPLORATION OF THE DATASET

Training Set

skimpy summary

Data Summary

dataframe	Values
Number of rows	6845
Number of columns	3

Data Types

Column Type	Count
int64	2
string	1

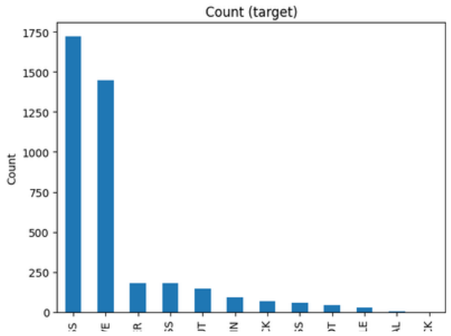
number

column_name	NA	NA %	mean	sd	p0	p25	p50	p75	p100	hist
label	0	0	5.408	3.014	0	2	6	8	11	
clip_duration	0	0	4.366	7.357	1	2	3	4	149	

string

column_name	NA	NA %	words per row	total words
clip_filename	0	0	6.7	46101

End



Validation set

skimpy summary

Data Summary

dataframe	Values
Number of rows	1603
Number of columns	3

Data Types

Column Type	Count
int64	2
string	1

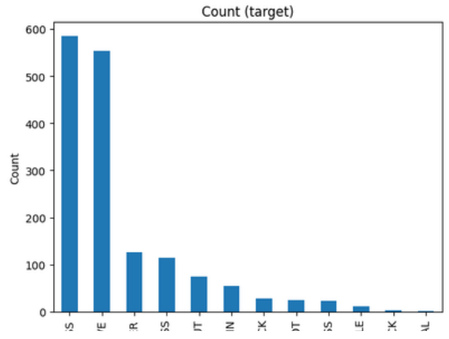
number

column_name	NA	NA %	mean	sd	p0	p25	p50	p75	p100	hist
label	0	0	5.379	2.987	0	2	6	8	11	
clip_duration	0	0	4.662	8.153	1	2	3	4	104	

string

column_name	NA	NA %	words per row	total words
clip_filename	0	0	7	11221

End



Test set

skimpy summary

Data Summary

dataframe	Values
Number of rows	3978
Number of columns	3

Data Types

Column Type	Count
int64	2
string	1

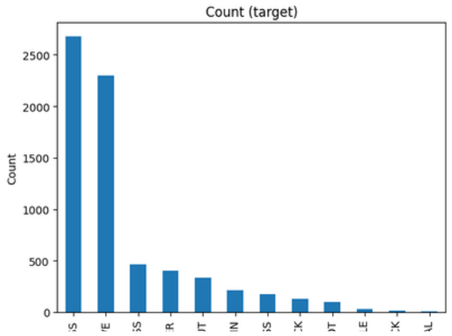
number

column_name	NA	NA %	mean	sd	p0	p25	p50	p75	p100	hist
label	0	0	5.404	3.019	0	2	7	8	11	
clip_duration	0	0	4	7.68	1	2	3	3	177	

string

column_name	NA	NA %	words per row	total words
clip_filename	0	0	5.9	23576

End



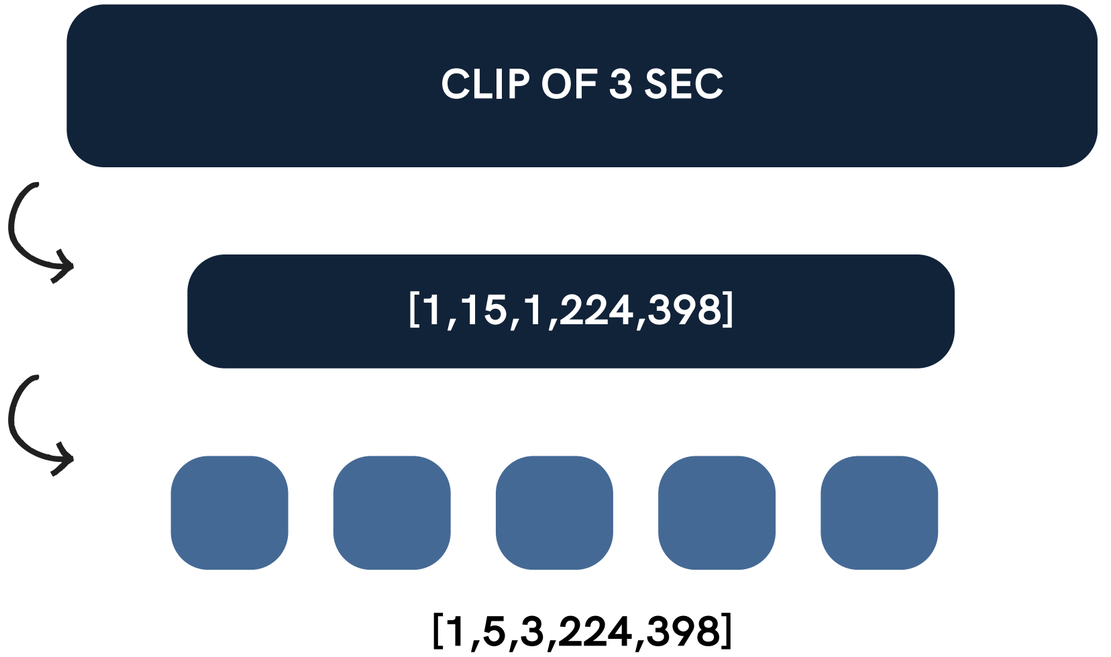
ActionRecognition Dataset class

The main idea is to take for each clip **15 frames**, one every 12 frames (because the mean length is 4), and then divide these frames into **stacks of 3 frames**.

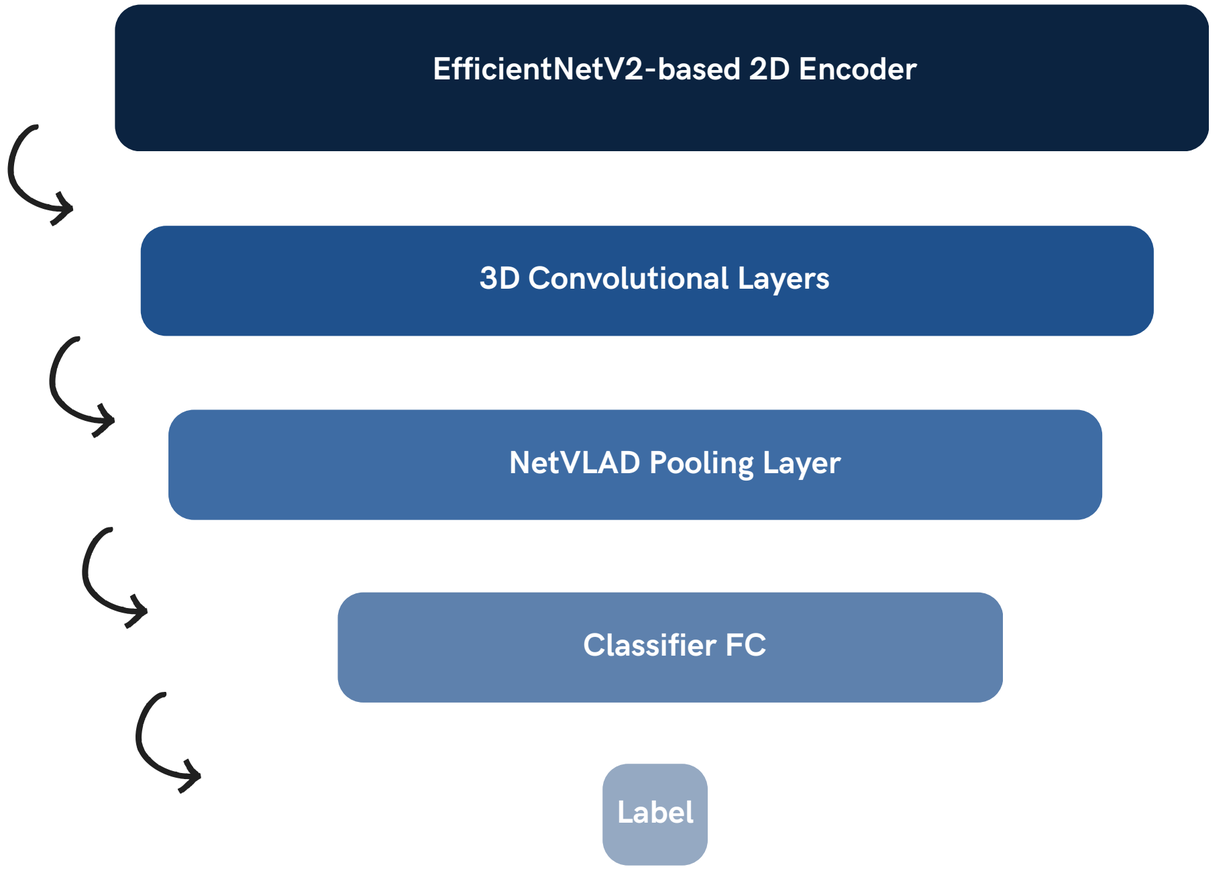
So in the end we will have a vector of **[1, 5, 3, 224, 398]** where **[B, T, C, H, W]**.

This will be useful to stack temporal information that will be processed by the network.

Videos will be transformed into **grayscale**



Network architecture



[1, 5, 3, 224, 398] [5, 192, 7, 13]	Input (B, T, C, H, W)
[1, 5, 192, 7, 13] [1, 192, 5, 7, 13]	Permute to [B, C, T, H, W] for 3D convolution
[1, 256, 5, 7, 13] [1, 1280, 7, 13]	Reshape for NetVLAD [B,T*C,H,W]
[1, 327680]	Input of the dense layer
[1, 12]	Output

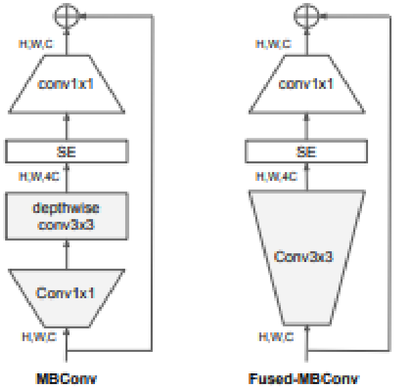
Submitted by Daniele Cecca

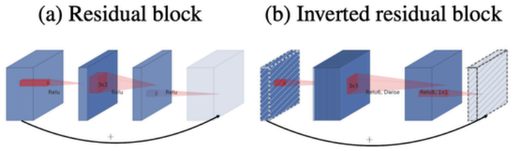
Building blocks

NETWORK

EfficientNetV2

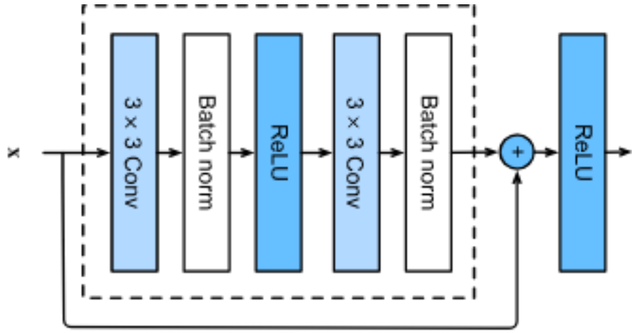
Inverted residual network

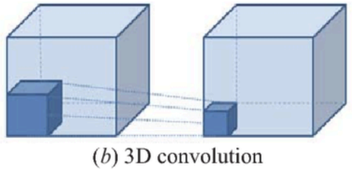




Residual block

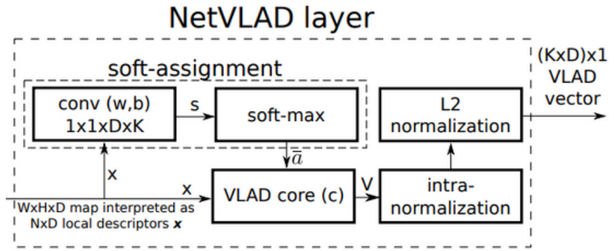
3D Convolution





(*b*) 3D convolution

NetVLAD



$$V(j,k) = \sum_{i=1}^{N} a_k(\mathbf{x}_i)\left(x_i(j) - c_k(j)\right),$$

$$V(j,k) = \sum_{i=1}^{N} \frac{e^{\mathbf{w}_k^T \mathbf{x}_i + b_k}}{\sum_{k'} e^{\mathbf{w}_{k'}^T \mathbf{x}_i + b_{k'}}} \left(x_i(j) - c_k(j)\right)$$

HyperParameters Tuning

NETWORK

I create a configuration that will be used by the **Weights and Biases agent** to set the different **hyperparameters** during various experiments. I used the simplest agent which chose the combination of the **parameters randomly**.

Parameters	Values
epochs	5
learning rate	[0.01, 0.001, 0.0001]
dropout	[0.4, 0.5]
Batch Size	[1,3,5]
loss function	focal loss
optimizer	adam

Focal Loss

NETWORK

Focal Loss is a loss function designed to address the challenge of class imbalance in tasks such as object detection.
Focal Loss is a modified version of the standard **Cross-Entropy Loss** that down-weights the contribution of easy-to-classify examples and focuses more on hard-to-classify examples. It is defined as

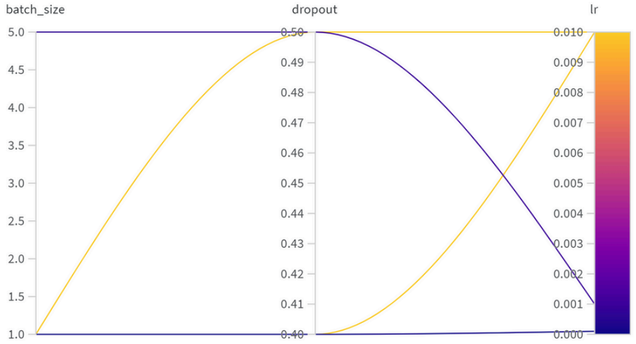
$$\mathrm{FL}(p_t) = -\alpha_t (1 - p_t)^{\gamma} \log(p_t)$$

- p_t is the model's estimated probability for the true class.
- α_t is a weighting factor for the class, balancing the importance of positive/negative example
- γ is a focusing parameter that adjusts the rate at which easy examples are down-weighted.

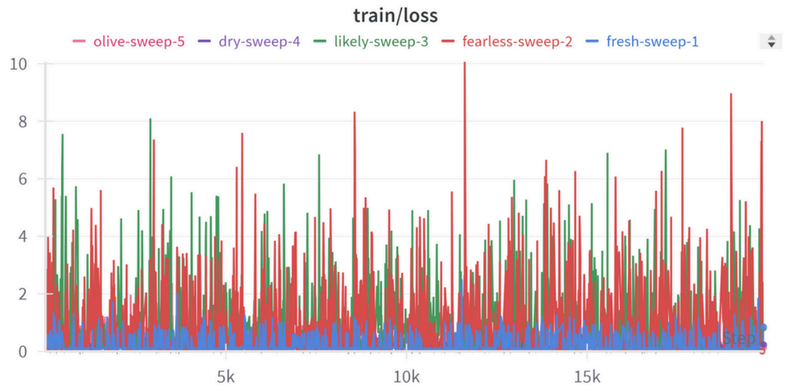
Submitted by Daniele Cecca

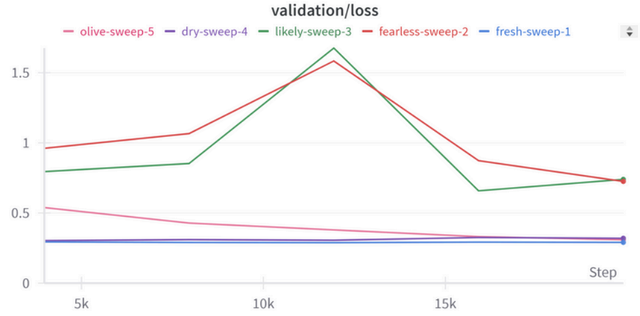
Training

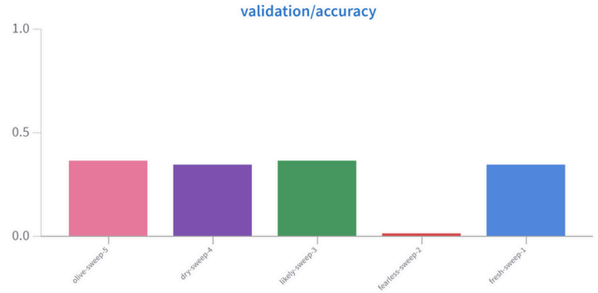
NETWORK



Parameter	Value
Batch Size	1
Drop Out	0.4
Epochs	5
Learning Rate	0.0001







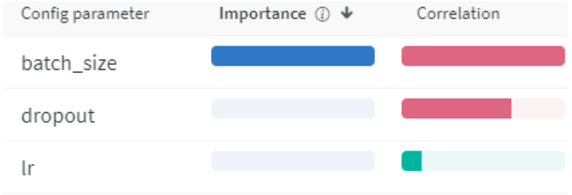
Training the best model

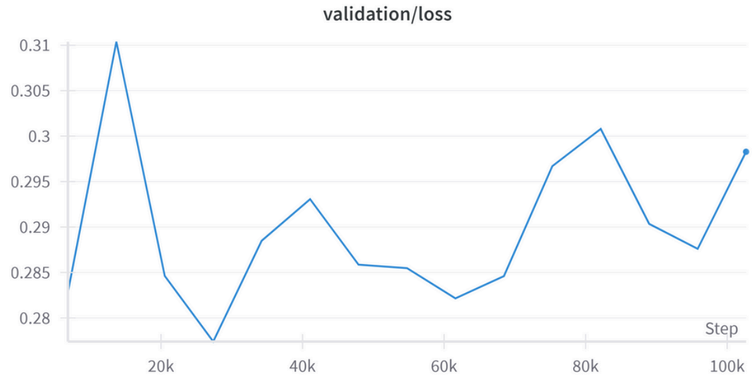
NETWORK

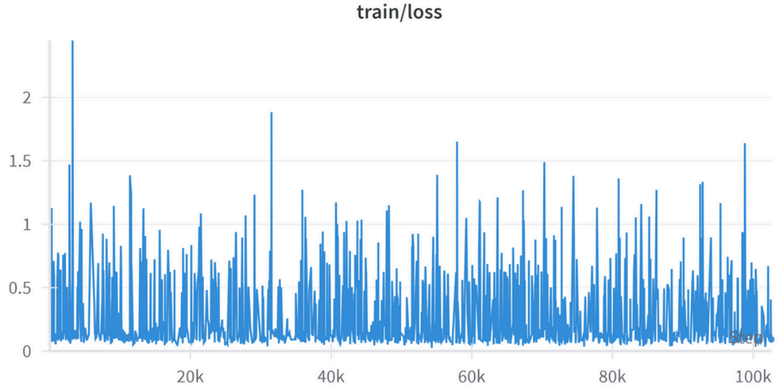
The behavior of the model is a little bit strange because both the **training loss** and the **validation** loss are **volatile.**

Despite it all I think the model is becoming better at generalizing because both validation and training losses are **decreasing on average**.

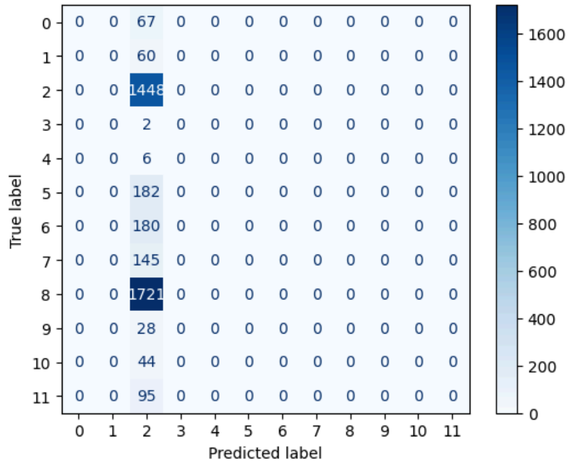
So in this case, it is difficult to talk about **overfitting** and **underfitting**.







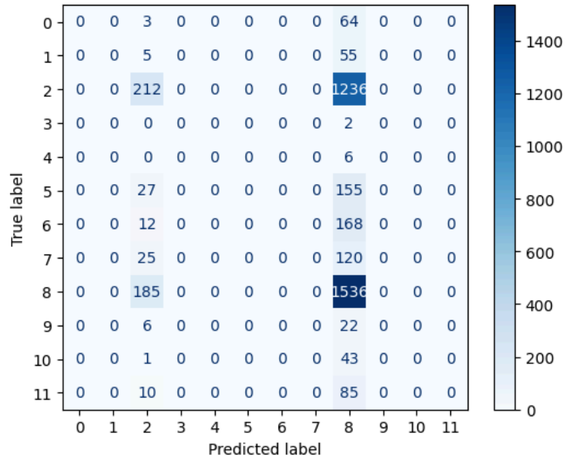
5 epochs model



- accuracy micro: 0.36
- precision micro: 0.36
- recall micro: 0.3
- f1 micro: 0.36

- accuracy macro: 0.36
- precision macro: 0.03
- recall macro: 0.08
- f1 macro: 0.045

20 epochs model



- accuracy macro : 0.44
- precision macro : 0.08
- recall macro : 0.09
- f1 macro: 0.07

- accuracy micro: 0.44
- precision micro: 0.44
- recall micro: 0.44
- f1 micro: 0.44

Evaluation

NETWORK

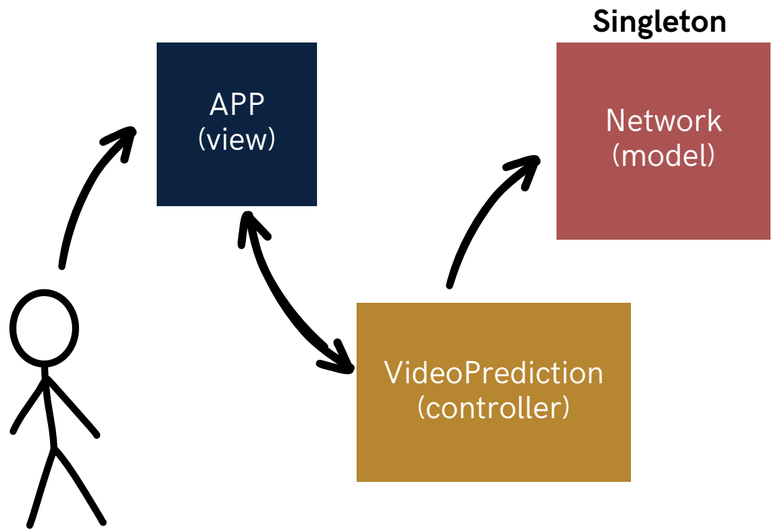
$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

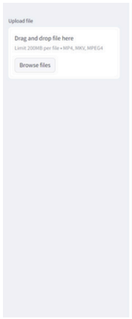
$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

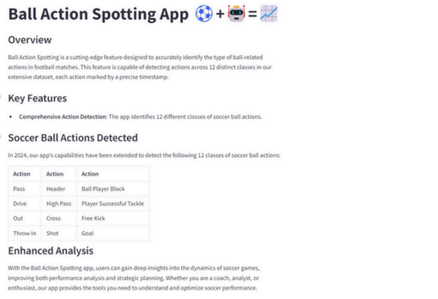
$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

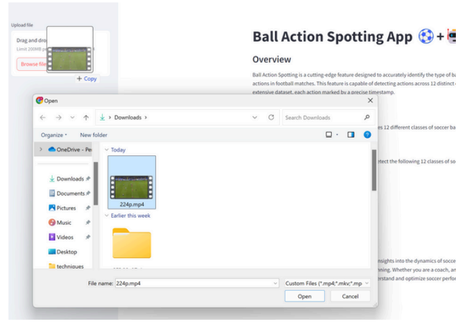
Streamlit app

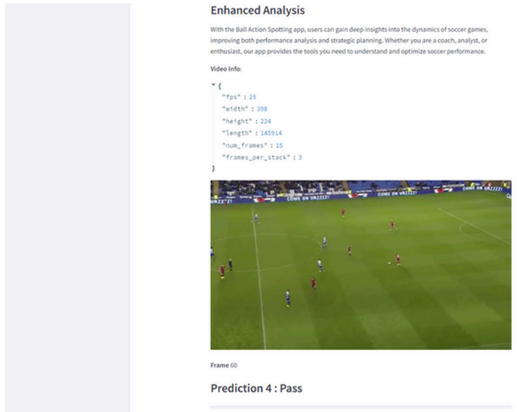




Load the file







Start classification

FUTURE WORKS

Network

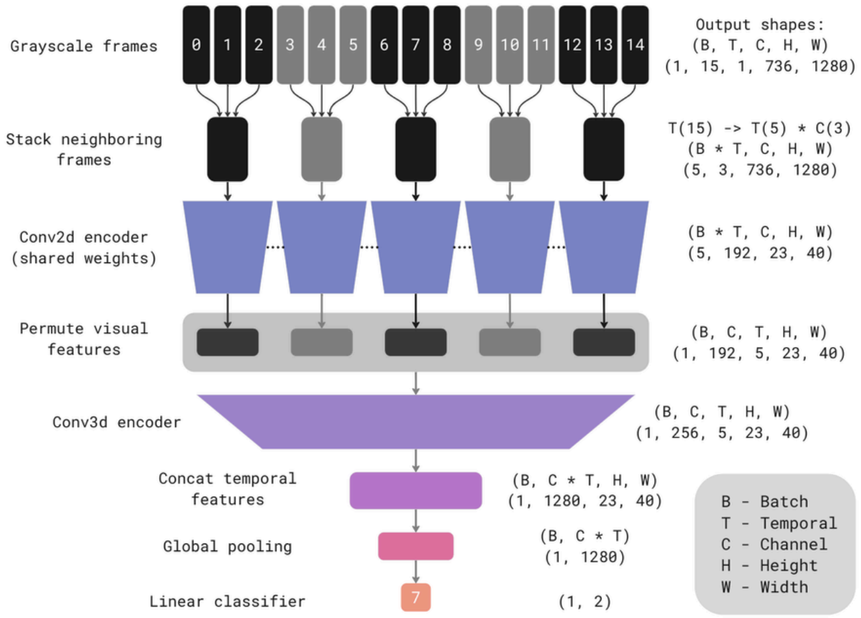
- Increase the dataset
- FineTune the model on the dataset of the task of action spotting
- Train for more epochs
- Try other hyperparameters
- Use high quality video
- Use RGB video
- Try other architectures

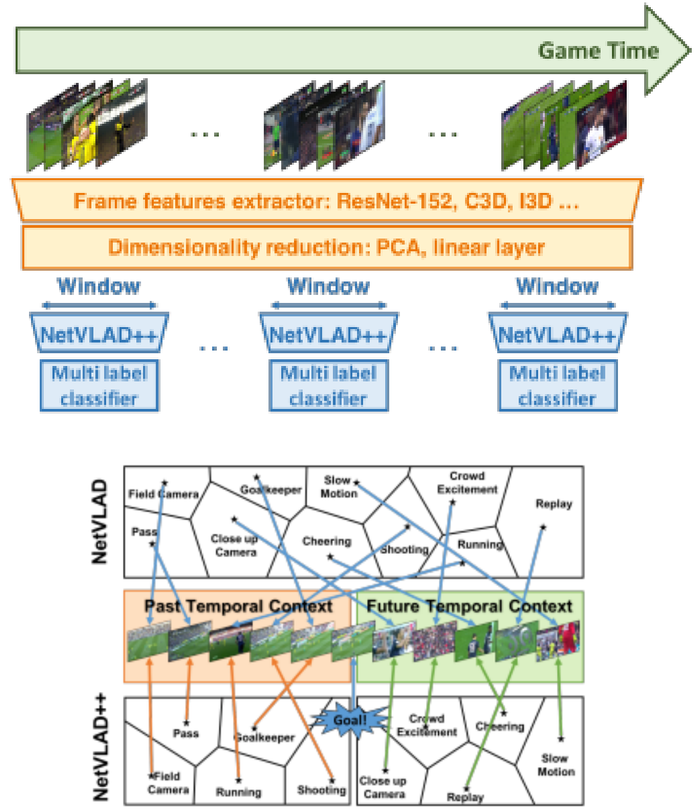
Streamlit

- Add a trigger to start the video when the classification begins.
- Use YOLO or another object detection model to assign the event to each player and compute some statistics relevant to the players.

State of art

INSPIRATION





Submitted by Daniele Cecca

State of art

INSPIRATION-BIBLIOGRAPHY

- NetVlad ++
- Encoder2D3DLateFusion
- EfficientNetV2
- SoccerNet
- NetVlad
- Vlad
- DatasetSoccerNet
- OverviewTrends
- InvertedResidualBlocks
- LibraryActionSpotting
- FocalLoss

THANKS FOR ATTENTION

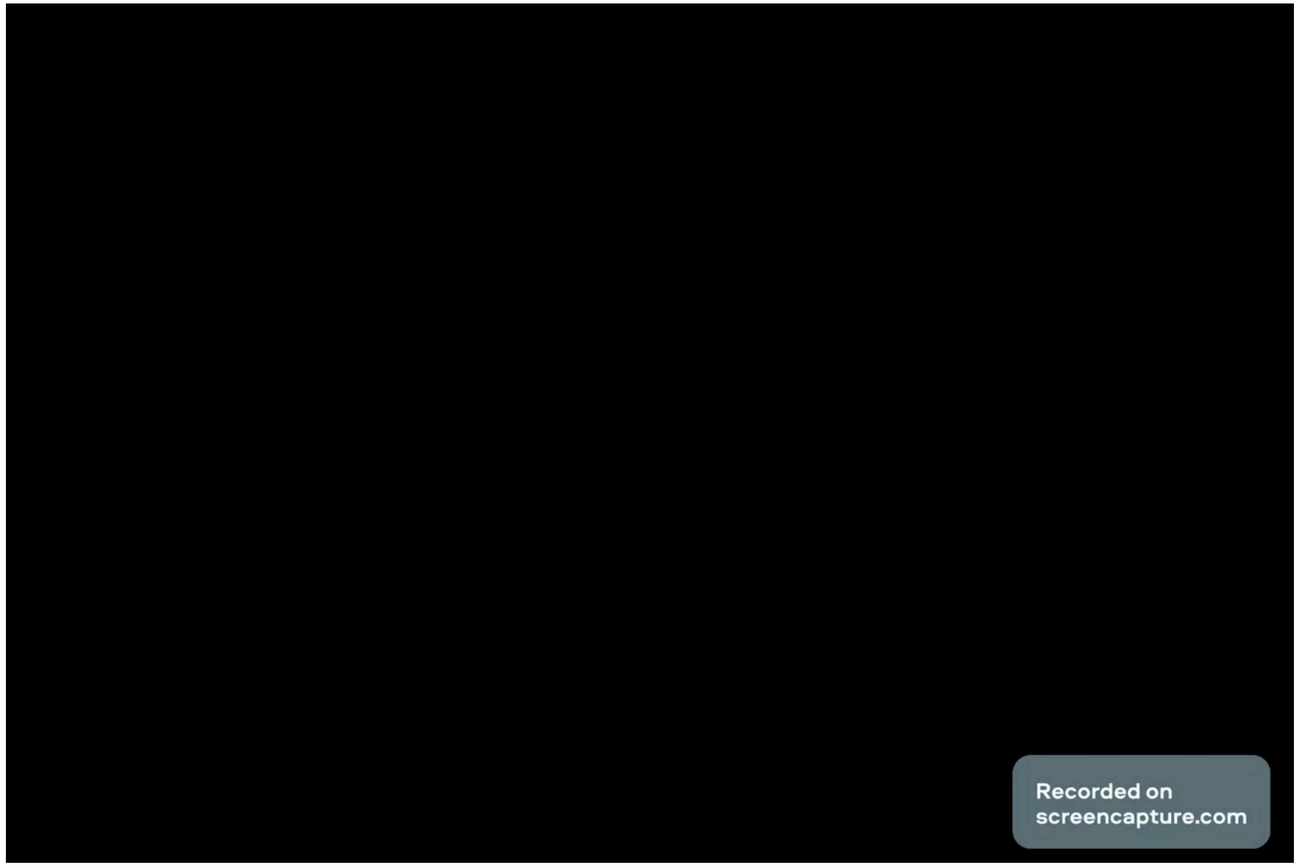
DANIELE CECCA MAT 914358

Advanced techniques
2023-2024

Presentation

Submitted by Daniele Cecca

DEMO



DEMO

DANIELE CECCA MAT 914358

Advanced techniques
2023-2024

Presentation