Soccer Events Detection

Federico Piscitelli 970949
Unimi

INTRODUCTION
BUILDING
TRAINING
RESULTS

Introduction

The aim of the project is to built an automatic system that is capable of distinguish between non soccer and soccer photos with the respective category of event

TACKLE

PENALTY

RED CARD

NO SOCCER

Dataset construction

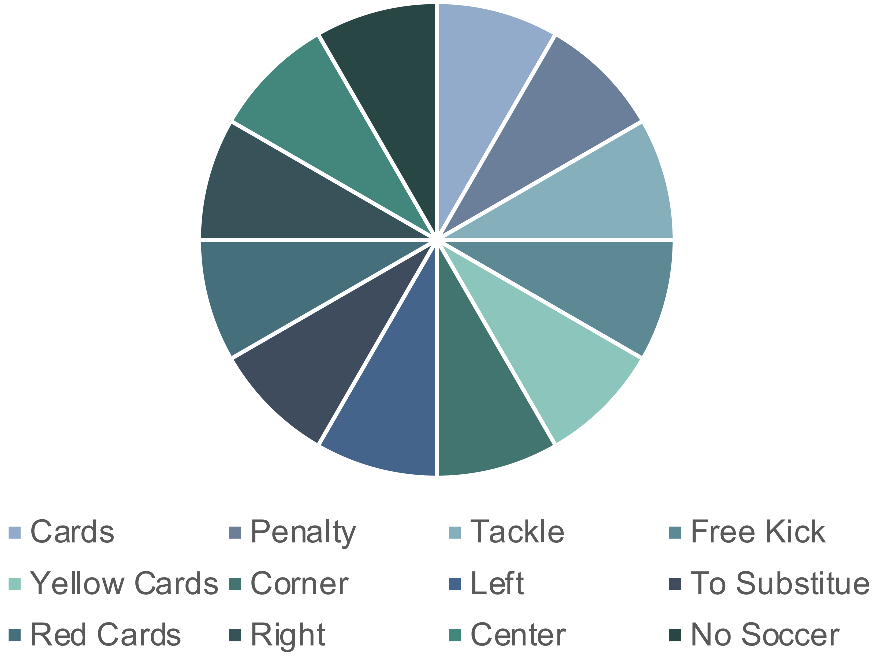
In order to be able to train the model over different image samples, a dataset was created.

1. **ANALYZE** existent SEV dataset
2. **FIND** a new dataset with general images
3. **INTEGRATE** general images with SEV dataset
4. **UPLOAD** the new dataset on Kaggle

Dataset details

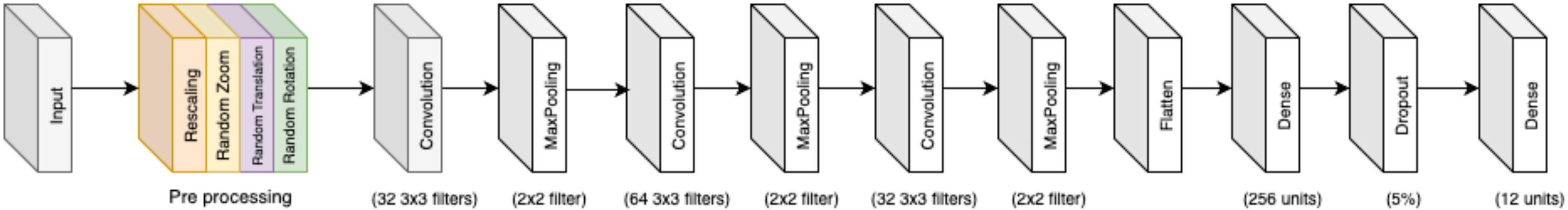
The dataset is splitted in 2 partitions:
- train
- test

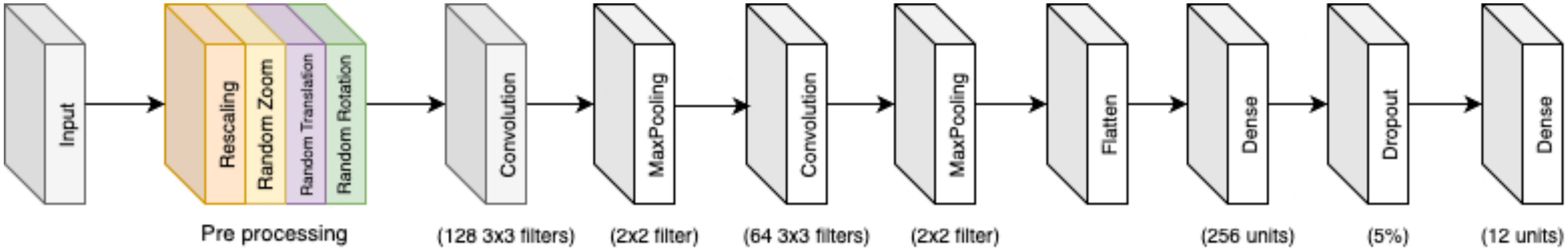
Each partition has 12 subset,
one for each class to identify.



CNN structures

Two CNNs were built to study the difference in performance due to the different number of layers and the different number of filters.





Pre-processing layers

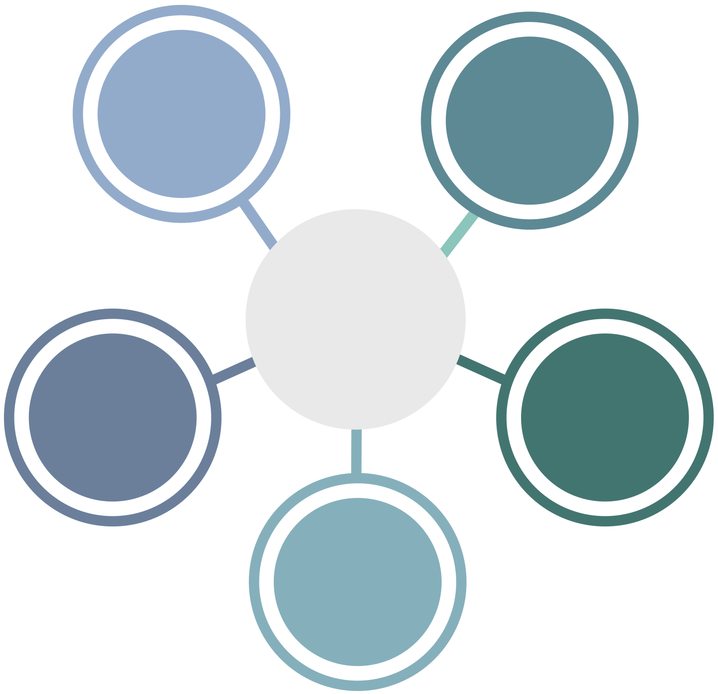
RESCALING	rescales the pixel values of the input images to a range between 0 and 1
RANDOM ZOOM	randomly applies zoom augmentation to the image
RANDOM TRANSLATION	randomly translates (shifts) the images horizontally and vertically
RANDOM ROTATION	apply a random rotation to the image

Other layers

CONVOLUTION
application of a sliding window function to a matrix of pixels representing an image

DROPOUT
randomly sets the output of 50% of the neurons to zero

DENSE
last layer of the convolutional neural network



MAXPOOLING
pull the most significant features from the convoluted matrix

FLATTEN
flattens the output from the previous layer into a 1D array

Training

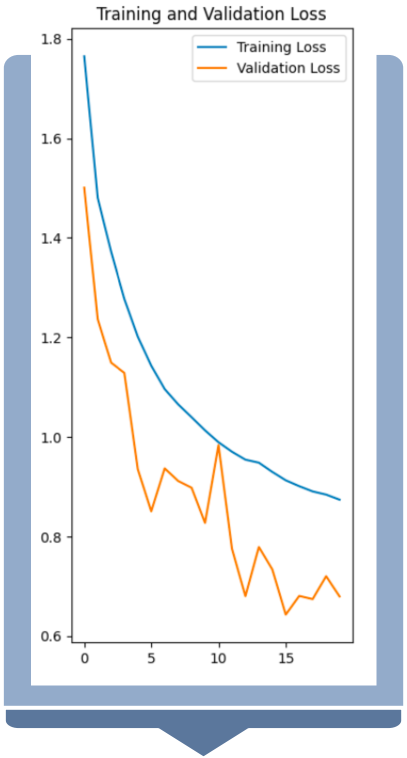
Both CNNs were trained over 20 epochs, during which they were provided with the valid images from the dataset.
To enhance training efficiency and model performance, several TensorFlow callbacks were implemented:

REDUCE LEARNING RATE	adjusts the learning rate if a certain metric fails to improve
EARLY STOPPING	halts the training process if the accuracy fails to increase for a specified number of consecutive epochs
MODEL CHECKPOINTS	saves the model's weights whenever an improvement is observed compared to the performance in a previous epoch

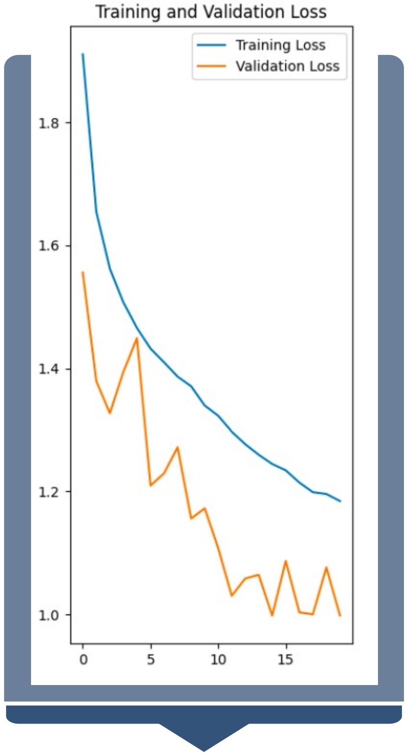
Training metrics

LOSS

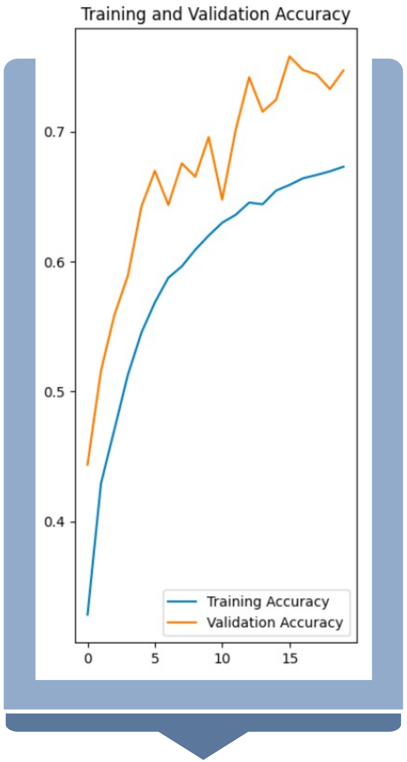
ACCURACY



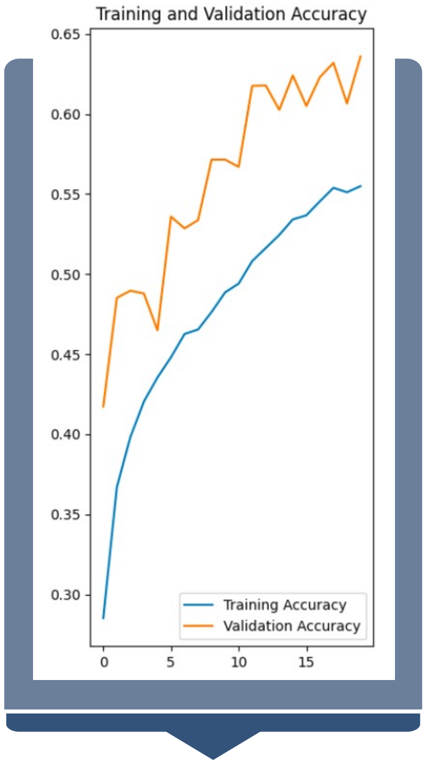
Model 1 ~ 0.7



Model 2 ~ 1.0

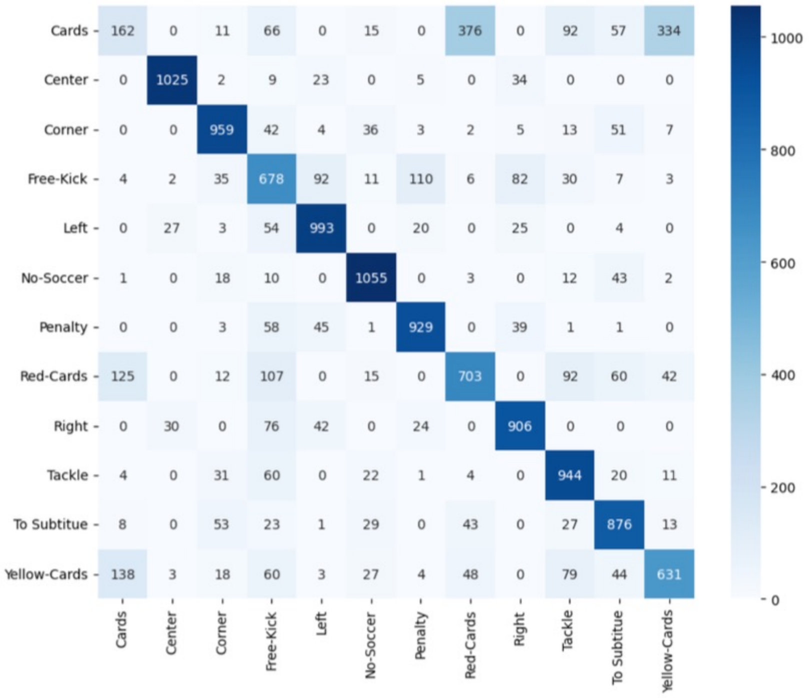


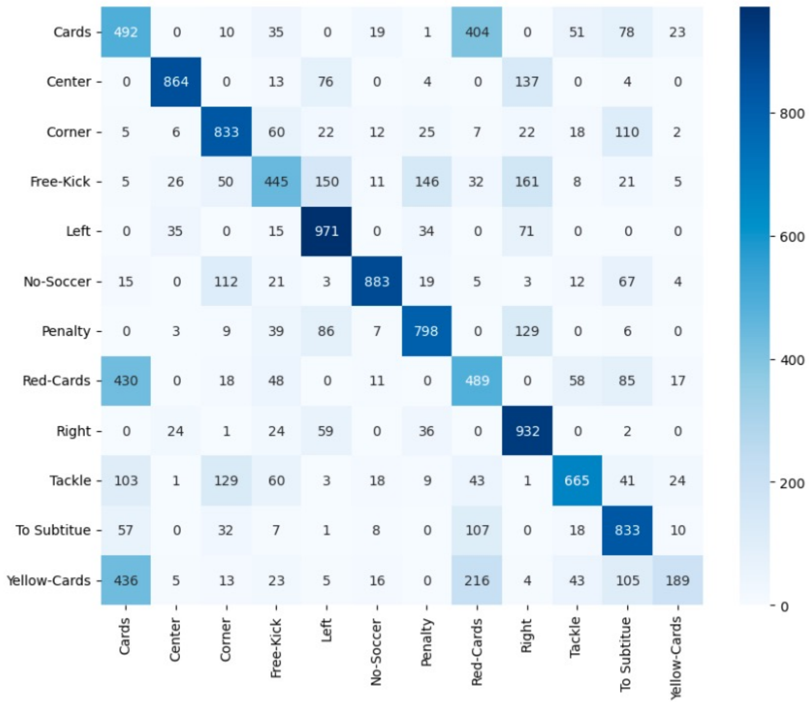
Model 1 ~ 0.74



Model 2 ~ 0.63

Confusion matrix



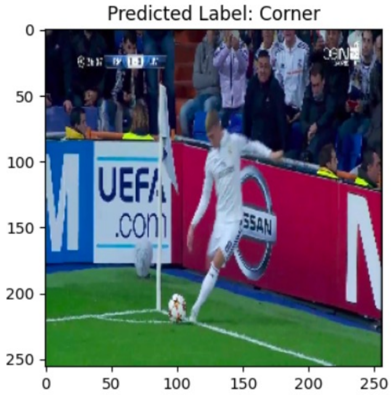


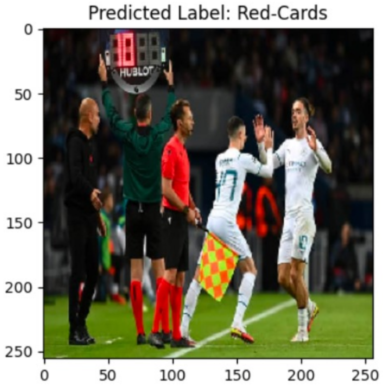
Final metrics

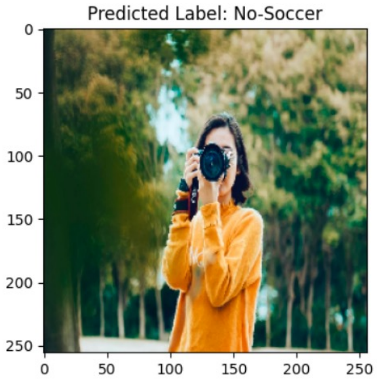
Model	Precision	Recall	Accuracy	F1-Score
Model 1	0.84	0.65	0.74	0.73
Model 2	0.81	0.47	0.63	0.62

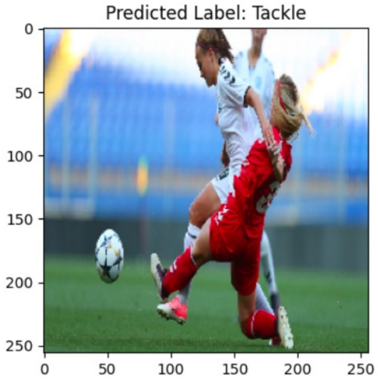
Class	Precision		Recall		F1-Score		Support
	Model 1	Model 2	Model 1	Model 2	Model 1	Model 2	
Cards	0.366516	0.318859	0.145553	0.442049	0.208360	0.370482	1113
Center	0.942962	0.896266	0.933515	0.786885	0.938215	0.838021	1098
Corner	0.837555	0.690141	0.854724	0.742424	0.846052	0.715328	1122
Free-Kick	0.545455	0.563291	0.639623	0.419811	0.588797	0.481081	1060
Left	0.825436	0.705669	0.881883	0.862345	0.852726	0.776179	1126
No-Soccer	0.871181	0.896447	0.922203	0.771853	0.895966	0.829497	1144
Penalty	0.847628	0.744403	0.862581	0.740947	0.855039	0.742671	1077
Red-Cards	0.593249	0.375288	0.608131	0.423010	0.600598	0.397723	1156
Right	0.830431	0.638356	0.840445	0.864564	0.835408	0.734437	1078
Tackle	0.731783	0.761741	0.860529	0.606199	0.790951	0.675127	1097
To Subtitue	0.753224	0.616124	0.816403	0.776328	0.783542	0.687010	1073
Yellow-Cards	0.604986	0.689781	0.598104	0.179147	0.601525	0.284424	1055

Prediction samples









Conclusion

The objective of the project, built a CNN capable of identifying footbal images, was achieved