# Soccer Event Detection
# Information Retrieval Project

Paolo Cortis

*Università Statale di Milano*
*Dipartimento di Informatica.*

**Abstract**

Event detection in video and image content is a relevant task which stands as a basis for many real world applications and innovative systems. The increasing importance of the task, together with its complexity, makes automated solutions all the more appealing. In this work various machine learning algorithms are applied to a soccer event detection task on the Soccer EVent (SEV) dataset. While performing experiments to understand how the analysis setup influenced the prediction performance of the methods, the focus was on defining a proper pipeline to solve the task, as well as comparing the different approaches and models' performances with appropriate figures of merit. After some considerations on the SEV dataset quality, the task was split into different phases, allowing for the experimentation of various approaches in each step. Convolutional Neural Networks (CNN) were deployed in different variations to handle the image content; techniques like transfer learning and data augmentation were also considered to improve models' performances and generalization capabilities. Furthermore, unsupervised clustering algorithms were studied to learn more about the nature of the task and provide an alternative approach.

## 1 Introduction

The ability to recognize human activities is one of the main subjects of study of computer vision and machine learning, and event detection serves diverse applications, ranging from video surveillance, human-computer interaction, activity recognition systems and robotics.

Unlike the traditional video-based action recognition approaches, this project investigates the usage of still images for the task, analysing whether the information extracted from this kind of data is relevant enough for machine learning models to exploit. Recent surveys [1], [2] present a detailed overview of the existing approaches in still image-based action recognition and investigate the different features and methods suited for this more complex approach. The major aspect being that still images do not carry any temporal information, thus, the prevailing spatio-temporal features for video-based action analysis, as well as the models' architectures exploiting the evolution of an action over time, like Recurrent Neural Networks, are not applicable in this context. Even if motion/video-based action recognition is still an active research field in computer vision, a study from 2009 [3] already presented the possibilities of action recognition on still images. In fact, many categories are found to be depicted unambiguously without motion or temporal information, and these actions can be recognized appropriately by machine learning models handling images. Another benefit of using this kind of data is that it is much simpler to acquire and store.

Inspired from these works, the project aims at defining an event detection methodology for soccer-related images, with the definition of a pipeline to tackle the problem while experimenting with different models and architectures, analysing their behavior, understanding and properly evaluating their performances.

# 2 Research question and methodology

## 2.1 Problem definition

Distinguishing the events of a match is an active research question in the soccer field, with various works aiming at the detection of events in a soccer game [4], [5]. The task has different relevant applications such as obtaining the statistics of events of the match, autonomously counting the number of relevant episodes (free kicks, fouls, tackles, etc.) and the summarization of a soccer match.

The project, inspired from a work of the University of Tehran [6], investigates various ways to tackle this multi-class image event detection task, by breaking down the problem in steps, defining an approach to reach the final classification and discussing the results obtained.

The primary goal of the system should be determining whether an input image depicts a specific event, classifying it into categories: red card, yellow card, corner, free-kick, penalty, tackle, substitution, center, left, right. Furthermore, generic soccer images, not pertinent with any particular event, should not be mistakenly categorized in one of the given classes. The system must also be able to handle random non-soccer related images.

## 2.2 System overview

A primary consideration to make is that, although a single model solution could be possible, dividing the problem in sub tasks and designing a pipeline consisting of multiple models is to be preferred. This *divide-et-impera* approach breaks down this articulated question in simpler and more general modules, allowing for cleaner models, more diverse and appropriate solutions for each step, as well as a more fine-grained optimization. The general process proposed to take on the problem, can be summarized as follows:

- Classification of soccer related images VS random non-soccer related images.
- Classification of the specific event in soccer related images + generic soccer class.
- Classification of card events in red card VS yellow card.

Fig. 1 presents a diagram of the whole system. This setup allows to clearly separate the problem in sub tasks, without any overlap. A first skimming isolates non soccer-related images, afterwards images are classified in events by a different module. Note that the red and yellow card events are identical, except for the card color, this makes their distinction complex in a single step; for this reason a specific classifier is arranged to handle this critical specification.

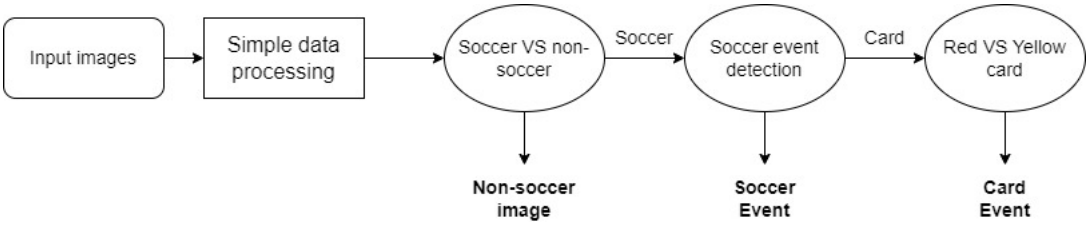


**Fig. 1** Data Processing Pipeline

## 2.3 Data processing and models' architectural choices

A preliminary phase consisted in the resize and color conversion of all images, resulting in 80x80 gray scale images for most sub tasks, while the RGB information was retained in specific instances, like for the classification of the card color. This step aims at aligning the dataset, speeding up training and reducing the memory requirements without affecting too much the final results. In addition, a basic form of data normalization was applied, proven to increase the performance and the training stability of models, resulting in pixel values ranging from 0 to 255 to be normalized in the 0-1 range.

The primary model employed in the project, serving as the baseline for task evaluation, is the Convolutional Neural Network (CNN). This architecture relies on convolutional operations, applying filters to the input to create feature maps that identify patterns and relevant features, making it well-suited for handling spatial information. The Relu activation function, Softmax output function, Adam optimizer and Categorical Cross-Entropy loss were used for this type of architecture. The CNN structure involves a sequence of convolutional layers followed by max pooling layers to downsample, retaining the most significant features. Multiple such blocks form the complete network. The convolutional results are then fed to a set of dense layers to generate the final predictions.

To avoid overfitting of the model (learning the training set up to the noise without being able to generalize over unseen data) the regularization technique of dropout was implemented. The idea is to partially limit the network capabilities during training by deactivating some neurons. This encourages the rest of the network to learn different aspects of the input, promoting robust feature extraction and generalization.

When designing a neural network, the choice of the architectural shape (number of layers, neurons, activation functions) and setting of learning hyperparameters (optimizer, batch size, learning rate) are critical for achieving high and reliable performances. At the state of the art, there is no unified method for finding the appropriate hyperparameters for a given task, and model selection is generally performed by relying on experience or involving empirical tests; these criteria were also applied for the definition of the models of this project. While automatic methods like Grid search or Bayesian optimization can be employed, these were not considered as out of the project scope, furthermore the scarcity of data did not allow for proper training of the meta models, resulting in suboptimal results with respect to hand made designs.

## 2.4 Classification of soccer VS non-soccer related images

This first step is formalized as a binary classification task to distinguish between soccer and non-soccer related images, the main goal being to filter out random images and keep only soccer related ones for the next modules. A first solution proposed to perform this decision was a simple Convolutional Neural Network. In this step the CNN architecture was composed of four convolutional blocks (the first three with 128 filters, the last one with 256 filters), with a flattening layer feeding the information to two dense layers with 128 neurons each with dropout.

Another option explored was clustering, an unsupervised technique to group by similarity the input data and divide the two categories (soccer, non-soccer), providing the advantage of the nonessential need of labeled training data. Given the nature of the task, clustering was deemed appropriate: while the random images cluster is definitely noisy and diverse, the soccer related images are similar and show consistent patterns that the model can exploit to group the data. These data characteristics, together with the fact that the number of clusters is known, led to the employment of two main clustering algorithms: K-Means and Agglomerative/Hierarchical clustering. K-Means divides the dataset into a specified number of clusters by randomly initializing each data point to one of the clusters and iteratively re-adjusting this decision to the cluster whose centroid is closest. Agglomerative clustering, instead, builds clusters hierarchically by iteratively merging the most similar clusters based on a similarity metric. These two techniques provide a good baseline to evaluate the application of clustering to the task, given that K-Means is simple and efficient, while Agglomerative

clustering is more flexible and capable of capturing more complex shapes and structures. The core aspect when applying these methods is to choose an appropriate feature space for the clustering algorithm to work on, the features have to be representative of the task at hand and be as correlated as possible with the class of the datapoints. An ideal feature space should present the two clusters as clearly separated, with an high intra-cluster distance and a low inter-cluster distance. The selected set of features included:

- Raw gray and RGB input image: the flattened pixel values.
- Features extracted by the convolutional layers of a CNN.
- Features extracted by the convolutional layers of a CNN condensed by a pooling layer.
- Texture related features.
- Histogram of Oriented Gradients (HOG).

The choice was motivated by the fact that the convolutional portion of a CNN is proven to be capable of extracting relevant features that even a simple clustering algorithm should be able to interpret with good results. The CNN used to extract the features was the pre-trained ResNet architecture, with the ImageNet task weights. The texture related features were extracted from the gray level co-occurrence matrix. This kind of matrix counts the number of times each pair of pixel values occurs in a window, yielding information regarding the visual texture of the image and can be probed for various texture-related features. In this case, the image was divided into small overlapping regions, for each window the co-occurrence matrix was computed and contrast, correlation, energy and homogeneity were extracted. As a last feature the HOG was considered: an histogram representing the gradient orientation in localized portions of the image, which provides insight on the objects and shapes present in the scene.

For all of the features in this list the Principal Component Analysis (PCA) technique was applied, resulting in eight more features. PCA is a statistical procedure used to decompose a multivariate dataset in a set of successive orthogonal components that exhibit the maximum amount of variance. The present data decomposition algorithm was applied in order to experiment with the same features reduced in two dimensions; this allowed to condense the information and to plot and visualize the feature space, granting the ability to determine its quality and inclination to the clustering task.

## 2.5 Soccer Event detection

The event detection task is formalized as a multi-class classification task where soccer images are assigned to the corresponding event. A first benchmark was established with a CNN composed of four convolutional blocks (two with 128 filters, two with 256 filters), with a flattening layer feeding the information to two dense layers with 128 neurons each with dropout.

In addition, considering also the nature of the dataset which will be touched upon later, the data augmentation technique was employed, with the intent to improve the generalization capabilities of the model. The goal of this procedure is to enlarge the dataset with pre-existing data by modifying a portion of the available data, in this case, changes in position, scale, rotation and brightness were considered, in order to to provide a more rich and diverse training base.

To study the task further, more experiments with transfer learning were carried out, making use of large pre-trained architectures for the extraction of relevant features from the images. In fact, the ResNet and VGG16 CNN architectures, proven to have great results in image classification tasks, were imported and tuned for the job at hand. The procedure involved "freezing" most of the convolutional layers of the networks, keeping their pre-trained weights on the ImageNet task intact for the feature extraction phase. The original prediction head was instead dropped and a dense layer was inserted after a global average pooling one, as well as an output layer, allowing fine tuning for the current task. This stands as a common procedure when applying transfer learning in the image field, and it is motivated by the observation that the first layers of a Convolutional Network extract more generic features (edges, corners, blobs) useful for many tasks, while the latter layers become instead progressively more specific to

the details of the classes in the original dataset. Since these architecture expect three channel input images, the chance was taken to experiment with both gray scale and RGB images, to determine whether the color information could be useful for this task, and to compare the two modalities.

### 2.5.1 Generic soccer detection

The next challenge to tackle was what the reference paper [6] defined as the *no highlight* problem, meaning the detection of soccer images not pertinent with any class. To correctly handle these generic soccer images a first intuition was to apply a threshold-based method on the prediction confidence of the event detection model. In such a scenario, the main model would output probabilities that could be interpreted as the likelihood of an image to belong to any of the classes. The idea was to classify as *generic soccer* the images whose maximum confidence score was below a certain threshold, meaning the model was unsure whether they were identifiable as a proper event. This solution did not perform as expected because of the inherent behavior of the classifier. Neural networks tend to be overly confident in their predictions, as reported in recent literature [7]. These models often struggle with out-of-distribution (OOD) samples that the network has not been exposed to during training, such as the generic soccer class in this case, and therefore are not handled correctly at test time. This naive solution utilizing the maximum Softmax probability for OOD detection, first applied in Hendrycks and Gimpel's work in 2016 [8], stands on the premise that OOD data should trigger relatively lower Softmax scores than that of in-distribution data. However, this is not the case as proven in these works and in the vast literature covering adversarial attacks [9], which presents state-of-the-art networks as overconfident, even when in great error. Plausible explanations include overfitting, causing the network to be excessively confident even with unrelated data, and the inherent nature of the Softmax activation function, which tends to magnify differences between class probabilities, increasing possible wrong biases in the network understanding.

A simpler alternative proved to function very well consisted in the inclusion of the generic soccer class in the training data, and the development of a CNN, similar to the previous ones, capable of understanding even the additional class.

## 2.6 Classification of Red vs Yellow card images

As previously mentioned, the red and yellow card classes differ only in the card color, while being two events with a very different impact on the soccer game, making the classification of these events much more critical. Hence, these classes were isolated and a specific classifier was developed for this fine-grained distinction. The CNN employed includes three convolutional blocks (two with 128 filters, one with 256 filters), with a flattening layer feeding the information to two dense layers with 128 neurons each with dropout, and, crucially, considered three channel RGB images.

# 3 Experimental results

## 3.1 Datasets

The Soccer EVent (SEV) dataset contains 60,000 soccer images collected in 10 categories, and was introduced in the reference paper [6]. The data was retrieved by crawling the Internet for unique images of different matches and by using the video frames of soccer games of the last few European leagues and extracting images related to the events. While the images have a good quality and are correctly labelled, upon a more detailed analysis it became evident that this dataset is rather inadequate. Because of the way in which most of the images where obtained, namely selecting video frames of soccer matches, the dataset is not that diverse, with many near duplicate samples. There are in fact numerous batches of 10-15 images that are subsequent frames of the video, with minimal differences. These near duplicate entries can

impact the models' abilities, introduce bias and skew the distribution of the data. This can cause numerous problems, such as overfitting, inflating the importance of certain observations during training, leading to poor generalization over new unseen data. Furthermore, duplicate entries can make the model appear more accurate than it actually is, distancing even the results on the test portion of the data from the actual generalization capabilities of the model; these considerations render this a crucial premise to make before reporting the results of the experiments. As a consequence of this analysis, the creation of an additional small set of data was deemed relevant, resulting in a 100 samples test dataset with 10 images for each class over which to test, even if in a very limited way, the generalization capabilities of the model on vastly diverse images depicting the same events.

When learning in a supervised manner the distinction between random images and soccer related ones, a dataset for the random portion is required, and, since it was not provided by the original paper authors, it was constructed for this purpose. To avoid a bias towards a particular format, noise or camera, the random class data was formed as a combination of diverse images from the random images dataset by Abhirup Ghosh [10] for a total of 1236 samples taken equally from all the classes, the dataset by Prasun Roy [11] for 64 more samples, plus one more batch of 100 images obtained from the Internet.

## 3.2 Experimental setup

All the presented CNN models were developed using Keras [12] with the TensorFlow backend, while the clustering algorithm were implemented with the Scikit-learn [13] library.

In order to train each model and evaluate its performance, the dataset was split into training and testing set. While the paper evaluated the architectures deployed over a single split of the SEV dataset, this project applied the Stratified Monte Carlo Holdout method to produce different arrangement of the train and test set, while keeping roughly the same class balance. This technique allows for the generation of multiple models, each with the same architecture, built and trained over different portions of the data (holdouts), the average performance of the models is considered as a statistically sound estimate of the overall performance of the architecture. For computational reasons, two holdouts were considered to train and evaluate each model, 20% of the whole dataset was used for testing, 80% for training. During the training phase each model went through the dataset for 100 epochs (although early stopping was set) with a batch size ranging from 10 to 32 samples.

When evaluating each model performance, a plethora of metrics are available, in this case, the Accuracy, the AUROC (Area Under Receiver Operating Characteristic) and the AUPRC (Area Under the Precision-Recall Curve) were deemed to be the most relevant. The Accuracy returns the number of correct predictions over all samples considered. The AUROC is the area under the curve where the x axis is the false positive rate and the y axis is the true positive rate. The AUPRC is the area under the curve where the x axis is the recall and the y axis is the precision. Furthermore, an historical report with the evolution of these metrics during the training procedure of each model, as well as the confusion matrix over the test set were considered to guide the development of models and to achieve a better evaluation of the models' behavior.

As for the assessment of the clustering algorithms, both K-Means and Agglomerative clustering were ran across all the previously mentioned features, providing insight on which algorithm handles the features best, as well as which features are more useful for the task. Given the availability of the ground truth, accuracy, AUROC, AUPRC, Mean squared error, Mutual information and Adjusted rand score were considered as extrinsic evaluation metrics of the clustering results. This was achieved by mapping clusters containing a majority of images labeled in a way to that label to compute these metrics. For an intrinsic evaluation the Silhouette score was considered, this metric combines the ideas of cluster cohesion and separation to determine the clustering quality without having to compare the results to the ground truth.

## 3.3 Results

### 3.3.1 Soccer VS non-soccer related images - CNN and clustering

For this classification module the CNN shows good results, with 0.963 accuracy, 0.983 AUROC and 0.978 AUPRC over the test set, indicating that the task can be dealt with comfortably employing a simple Convolutional Neural Network. Still, for the aforementioned reasons, this model, like most of the others, was also evaluated on a different test dataset with vastly different characteristics from the SEV dataset, resulting in a considerable drop in performances setting the accuracy metric to 0.718, the AUROC to 0.800 and the AUPRC 0.217. This indicates that some overfitting was indeed present due to the dataset quality, but the results are still acceptable, proving that the model can still generalize over distinct unseen images.

As for the second approach to this sub task, Tables 1-2 provide a comprehensive view including the results of the two clustering algorithms with respect to the feature used as input. The features that produced the better performances are RGB raw images and, especially, the features extracted by the CNN layers of RGB images, in particular when those were condensed by a pooling layer. These inputs led to a much better clustering and, more precisely, generated the most indicative feature space with respect to the task, providing a good separation of the data that could be exploited by the algorithms, as reported by the considerable increase in performances.

This result further proves the effectiveness of Convolutional layers when processing image content; in comparison, the texture related features of the co-occurrence matrix and the histogram of oriented gradients led to much lower results. This examination is further supported by the visualization of the PCA reduced features; in particular Fig. 2 shows how the space of the RGB features extracted from CNN is much more informative for the task than that of texture features, and the consequent improvements when applying Agglomerative clustering to those two spaces. The application of PCA to the space allowed for these visualization but didn't change much in terms of results compared to the non-reduced counterpart.

Another consideration is that color information really helped in the definition of an informative feature space; this seems reasonable as the color palette of a soccer event is similar inter-cluster, while being much different with respect to non-soccer related images (intra-cluster).

**Table 1** K-Means performance on features for Soccer vs non-soccer task

| Feature | Accuracy | AUROC | AUPRC | Mif | Ars | Silhouette | MSE |
|---|---|---|---|---|---|---|---|
| Gray Raw Image | 0.595 | 0.597 | 0.519 | 0.019 | 0.036 | 0.060 | 0.405 |
| Gray CNN Features | 0.638 | 0.620 | 0.551 | 0.038 | 0.075 | 0.119 | 0.362 |
| Gray CNN Features + Pooling | 0.628 | 0.612 | 0.541 | 0.031 | 0.064 | 0.104 | 0.372 |
| RGB Raw Image | 0.665 | 0.677 | 0.575 | 0.070 | 0.108 | 0.164 | 0.335 |
| RGB CNN Features | **0.865** | **0.859** | **0.816** | **0.302** | **0.533** | **0.617** | **0.135** |
| RGB CNN Features + Pooling | **0.894** | **0.892** | **0.843** | **0.352** | **0.620** | **0.694** | **0.106** |
| Matrix Features | 0.539 | 0.524 | 0.475 | 0.001 | 0.005 | 0.010 | 0.461 |
| HoG Features | 0.561 | 0.581 | 0.506 | 0.018 | 0.013 | 0.016 | 0.439 |

**Table 2** Agglomerative clustering performance on features for Soccer vs non-soccer task

| Feature | Accuracy | AUROC | AUPRC | Mif | Ars | Silhouette | MSE |
|---|---|---|---|---|---|---|---|
| Gray Raw Image | 0.600 | 0.592 | 0.519 | 0.018 | 0.039 | 0.066 | 0.400 |
| Gray CNN Features | 0.648 | 0.628 | 0.563 | 0.047 | 0.086 | 0.133 | 0.352 |
| Gray CNN Features + Pooling | 0.648 | 0.628 | 0.563 | 0.047 | 0.085 | 0.132 | 0.352 |
| RGB Raw Image | 0.571 | 0.565 | 0.500 | 0.009 | 0.020 | 0.036 | 0.428 |
| RGB CNN Features | **0.849** | **0.856** | **0.760** | **0.294** | **0.489** | **0.580** | **0.150** |
| RGB CNN Features + Pooling | **0.869** | **0.871** | **0.793** | **0.308** | **0.544** | **0.628** | **0.131** |
| Matrix Features | 0.539 | 0.509 | 0.466 | 0.001 | 0.002 | 0.009 | 0.461 |
| HoG Features | 0.563 | 0.594 | 0.514 | 0.056 | 0.012 | -0.078 | 0.437 |

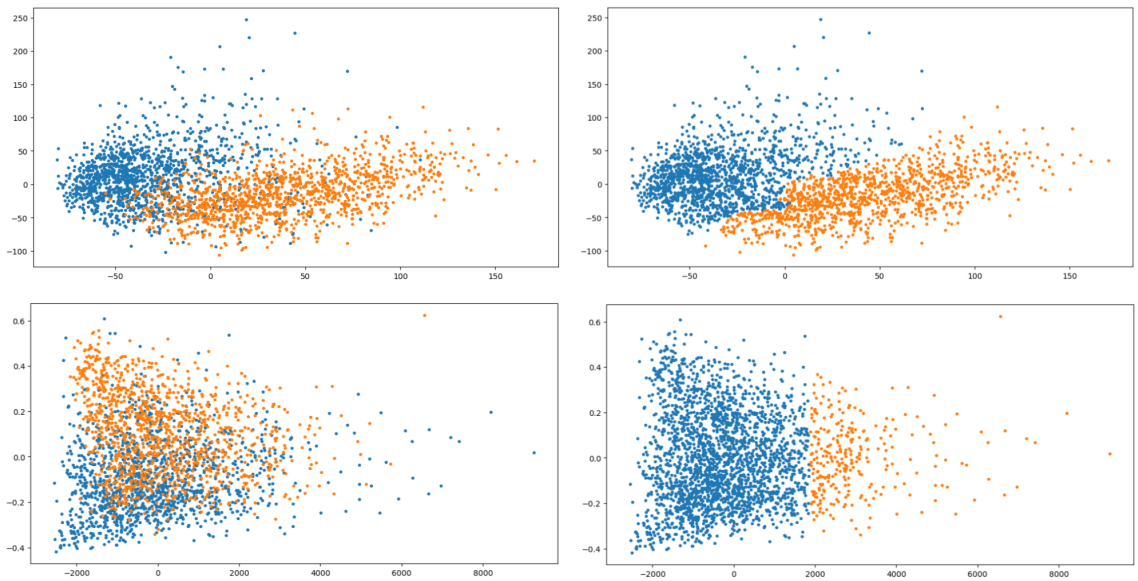

**Fig. 2** PCA feature space of RGB CNN and texture features (left) and Agglomerative clustering results on the same spaces (right)

In terms of evaluating K-Means compared to Agglomerative clustering, the latter proved to be more flexible and aligned itself better with the original cluster shape, this is especially evident when looking at the PCA visualizations; that said, the results of the two algorithms are comparable.

The performance of both algorithm, while satisfying, is still lower with respect to that of the CNN, but with the major benefit of being unsupervised methods.

### 3.3.2 Soccer event detection

Table 3 provides a comprehensive view of all the model results for the event detection task, comparing test accuracy, AUROC and AUPRC metrics on the SEV dataset with the same metrics on the 100 samples test dataset, while Fig. 3 presents a barplot displaying the test accuracy of all models on the SEV dataset (test AUROC and AUPRC display a similar trend).

The baseline CNN shows very good results, proving that even a simple convolutional model can handle the task well. However, because of the possibility of very similar images being present in both train and test set, the model shows clear signs of overfitting, manifesting a considerable drop in performances on the test dataset with respect to SEV data (-25.59% test accuracy, -17.00% test AUROC, -44.02% test AUPRC). In any case, the results are still acceptable, proving that some generalization capabilities are retained, even if not at the levels suggested by the test portion of SEV.

When implementing data augmentation, the model performances didn't change much on the SEV dataset; however the results improved in the test dataset with an increase of +1.60% in test accuracy, +0.73% in test AUROC, +6.31% in test AUPRC, proving that the technique did indeed slightly favor the generalization capabilities of the model.

Transfer learning didn't perform as well compared to these simpler CNN architectures with fewer parameters. This result is not uncommon and can be attributed to several factors, like the nature of the task and the models' architectures. It is likely that, to correctly classify soccer events, a set of features different from those extracted by ResNet and VGG16 is required, hence the simpler architecture, trained from scratch, is capable to extract the more relevant ones and align itself better with the task. Furthermore, these architectures are trained on a huge and diverse dataset, whereas in this case images are much more similar and the distinction between classes is not so pronounced. An additional possible cause for this is overfitting, in fact, these pre-trained models have many parameters, and fine-tuning them on a small

**Table 3** Soccer model performance comparison

| Model | SEV Accuracy | SEV AUROC | SEV AUPRC | dataset Accuracy | dataset AUROC | dataset AUPRC |
|---|---|---|---|---|---|---|
| Basic CNN | 0.926 | 0.994 | 0.975 | 0.689 | 0.825 | 0.539 |
| CNN + Data Augmentation | 0.905 | 0.992 | 0.963 | 0.700 | 0.831 | 0.573 |
| Tuned ResNet gray | 0.786 | 0.974 | 0.863 | 0.544 | 0.743 | 0.399 |
| Tuned ResNet RGB | 0.757 | 0.971 | 0.845 | - | - | - |
| Tuned VGG16 gray | 0.870 | 0.988 | 0.941 | 0.700 | 0.831 | 0.553 |
| Tuned VGG16 RGB | 0.901 | 0.990 | 0.956 | - | - | - |
| Basic CNN + generic class | 0.933 | 0.994 | 0.975 | 0.620 | 0.789 | 0.457 |
| Cards CNN | 0.956 | 0.990 | 0.989 | - | - | - |

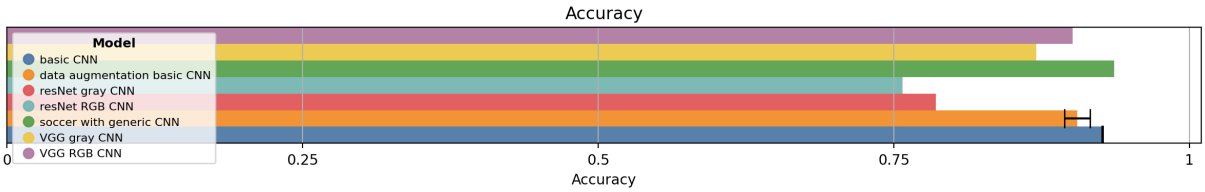


**Fig. 3** Event Detection Model Accuracy Comparison

dataset can lead to poor generalization capabilities. Still, results are acceptable, and, compared to ResNet, the VGG16 architecture performs much better on the task with an average increase of about +14.79% in test accuracy, +1.70% in test AUROC, +11.07% in test AUPRC. Another interesting result is that, while the ResNet doesn't show signs of improvement when handling RGB images with respect to gray scale ones, VGG16 improved in performances with an increase of +3.56% in test accuracy, +0.20% in test AUROC, +1.59% in test AUPRC. This indicates that the architecture finds the color information somewhat useful to perform the classification, even if it's not that decisive; this is logical as the colors of the shirts or that of the field do not help in distinguishing a penalty with respect to a tackle for example. When comparing these architectures to the baseline CNN on the test dataset, the results don't indicate any particular fluctuations in performances, and are in line with the outcomes on the SEV dataset, suggesting that these larger networks were still capable of retaining an acceptable generalization capability, especially the tuned VGG16.

Given these results, the model deployed for the task when considering also the generic soccer class was based on the baseline CNN. This final architecture for the event detection task showed the best results on the SEV dataset, with only a slight decrease on the test dataset, proving capable of detecting all the previous classes, as well as the generic soccer class.

As for the final classification step, the CNN employed to distinguish red and yellow cards displays good results, proving that, when handling RGB images, this task can be dealt with quite smoothly.

# 4 Conclusion

After analysing the complexity of the task and the dataset quality, different methods and features were considered for the soccer event detection task on the SEV data and also evaluated on a more diverse test dataset. The proposed approach split the task in a three-step pipeline to handle non soccer images, soccer events and red/yellow cards. Convolutional networks proved capable of handling most of the process, especially when paired with the data augmentation technique. Unsupervised clustering techniques were also considered and achieved good results when handling a good feature space.

While tuned ResNet and VGG16 didn't increase performances, a possible next step could be to experiment further with other types of architectures, considering for example Vision Transformers, which would focus on different aspects of the images by processing the global input with the attention mechanism, leading to a different feature space that could potentially improve results.

# References


[1] Guodong Guo, Alice Lai. A survey on still image based human action recognition. Lane Department of Computer Science and Electrical Engineering, West Virginia University. Pattern Recognition, 2014.

[2] Maryam Ziaeefard, Robert Bergevin. Semantic human activity recognition: A literature review. Computer Vision and Systems Laboratory, Department of Electrical and Computer Engineering, Laval University. Pattern Recognition, 2015.

[3] Abhinav Gupta, Aniruddha Kembhavi, Larry S. Davis. Observing Human-Object Interactions: Using Spatial and Functional Compatibility for Recognition. University of Maryland-College Park, Department of Computer Science. IEEE Transactions on pattern analysis and machine intelligence, Vol. 31, No. XX, 2009.

[4] Y. Hong, C. Ling, and Z. Ye. End-to-end soccer video scene and event classification with deep transfer learning. International Conference on Intelligent Systems and Computer Vision (ISCV). IEEE, 2018, pp. 1–4.

[5] M. Z. Khan, S. Saleem, M. A. Hassan, and M. U. G. Khan. Learning deep c3d features for soccer video event detection. 14th International Conference on Emerging Technologies (ICET). IEEE, 2018, pp. 1–6.

[6] Ali Karimi, Ramin Toosi, Mohammad Ali Akhaee. Soccer Event Detection Using Deep Learning. School of Electrical and Computer Engineering, College of Engineering, University of Tehran. arXiv:2102.04331v1, 2021.

[7] Hongxin Wei, Renchunzi Xie, Hao Cheng, Lei Feng, Bo An, Yixuan Li. Mitigating Neural Network Overconfidence with Logit Normalization. Proceedings of the 39th International Conference on Machine Learning. arXiv:2205.09310v2, 2022.

[8] Hendrycks, D. and Gimpel, K. A baseline for detecting misclassified and out-of-distribution examples in neural networks. arXiv:1610.02136, 2016.

[9] Anh Nguyen, Jason Yosinski, Jeff Clune. Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images. arXiv:1412.1897, 2015

[10] https://www.kaggle.com/datasets/ezzzio/random-images

[11] https://www.kaggle.com/datasets/prasunroy/natural-images?resource=download

[12] Francois Chollet et al. Keras. https://github.com/fchollet/keras, 2018.

[13] Pedregosa, F. and Varoquaux, G. and Gramfort, A. and Michel, V. and Thirion, B. and Grisel, O. and Blondel, M. and Prettenhofer, P. and Weiss, R. and Dubourg, V. and Vanderplas, J. and Passos, A. and Cournapeau, D. and Brucher, M. and Perrot, M. and Duchesnay, E. Journal of Machine Learning Research, Vol.12, pp. 2825-2830, 2011.