EIT Digital Masters - M1 DSC

Research Report

Detection of actions in Sports Videos using Multiple Instance Learning

Supervisors:
Pr. Frederic Precioso, Melissa Sanabria Rosas

Author:
Sherly

Abstract

There has been many research work on sports data in the recent years but there has yet been many studies on issues raised from sports videos that are highly homogenous. Homogeneity has made it difficult to classify highly similar events from segments of videos into different classes. The dataset studied in this project are videos of football games. Many parts of the video segments contain highly similar events that belongs to different classes i.e. a pass can be of importance or not. As such, in this project, we study the application of Multiple Instance Learning algorithm on this kind of data where an event is viewed as part of a bag of events. The challenge is to find the right application of the algorithm on this dataset that enables us to classify these highly homogeneous video data into the right classes.

Key words: Sports, Multiple Instance Learning, Neural Networks

Contents

1 Introduction

1.1 Context

In the recent years, several approaches have shown promising results to detect, extract and classify activities in video datasets like ActivityNet or THUMOS-14. Often, these datasets contain actions that are clearly different like Diving and Billiards. However, in the case of other sports, we could often find actions that are highly homogeneous within the video causing it to be more challenging for us to differentiate.

In this project, the dataset consists of videos of football games with the time intervals of different actions like *goal on field, free-kick, saved field, yellow card,* etc as well as metadata. Unfortunately, many parts of these intervals contain events (i.e. sub-sequences) that also belongs to the background class where the section of the video has no actions. For instance, a *pass* or a *throw-in* can be present whether inside a segment labeled as *goal on field* or in some part of the video where nothing important is happening. Therefore, it makes it more difficult to train a Machine Learning algorithm that would perform well due to the confusion in the algorithm where very similar segments belong to different classes.

1.2 Motivations

Multiple Instance Learning (MIL) is one of the potential solution to the problem mentioned. In this type of learning, we group the instances in bags (or sets). A bag is labeled as *negative* if all the instances in the set are indeed negative. On the other hand, a bag is labeled as positive if there is at least one instance in the set which is positive.

For this problem, instances would be all the small events like *pass*, *throw-in, goal, tackle, card, miss,* etc. and the bags are the segment actions. There are also complexities in the definition of instances where an instance could be defined as snippets of the videos with fixed frames per second or the full duration of the segments. Negative bags are the segments labeled as Background because we know in this segments there are not events that represent an action, and the positive bags are the segments labeled with actions (free-kick, saved field, etc).

1.3 Goals

The goal of this project is train a multiple instance learning based neural network to achieve better accuracy of classifying similar segments. We would like to find an implementation of Multiple Instance Learning that can handle the video features of our dataset.

2 Background

This section describes the concepts and theory that is necessary to understand the work that is done in this project. It will first describe the concept of Multiple Instance Learning (MIL). the application of classification in the context of sports actions and evaluation metrics.

2.1 Multiple Instance Learning

Multiple Instance Learning is a type of supervised learning. [1]Traditionally, a learning algorithm receives a set of instances with its labels individually, however, in this case, the learning algorithm receives sets of instances where it each sets carries a single label. We refer to these sets of instances as bags. In the context of our project, sequences of actions, where actions are highly similar may be classed differently and thus the traditional model of labelling instances individually fails. Only a collection of instances, forming a sequence of actions could be labelled. Thus, Multiple Instance Learning formalizes this problem. In this learning, we receive a set of bags, each of which is labeled positive or negative. Each bag contains many instances, where each instance is a point in feature space. A bag is labeled negative if all the instances in it are negative. On the other hand, a bag is labeled positive if there is at least one instance in it which is positive. From a collection of labeled bags, the learner tries to induce a concept that will label unseen bags correctly.

Multiple instance learning (MIL) was originally proposed for drug activity prediction[1]. It is currently applied over many domains such as the multimedia data. A text can be decomposed into instances of sentences of paragraphs and an image can be decomposed into multiple regions in the multiple instance learning application.

[1] https://en.wikipedia.org/wiki/Multiple_instance_learning

There are many algorithms have been proposed to solve the MIL problem. According to the survey by Amores [2], there are three paradigms to MIL algorithms: instance-space paradigm, bag-space paradigm and embedded-space paradigm. Instance-space paradigm learns instance classifier and aggregates instance level classifiers to give bag classification. Bag-space paradigm treats bag as a whole and by computing bag-to-bag distance nearest neighbor or Bayesian classifier is performed to obtain bag classification. Embedded-space paradigm translates a bag into a feature space for a compact representation for the bag and by applying classical classifiers on it, we obtain the bag classification.

2.1.1. Multiple Instance Neural Networks

In this project, we will be applying Multiple Instance Neural Networks with architectures proposed by Wang et al, 2018 [3]. A Multiple Instance Neural Networks takes a various number of instances as input. For each instance, its representation is gradually learned layer by layer guided by multiple instance supervision. There are two networks proposed by Wang [3], mi-Net which works in instance-space paradigm and MI-Net which works in embedded space paradigm. In our explorations for this project, we will only be looking at MI-Net architectures which performs bag level predictions, more specifically the MI-Net and MI-Net with Deep Supervision. The two architectures are shown in Figure 1 and Figure 2 below.

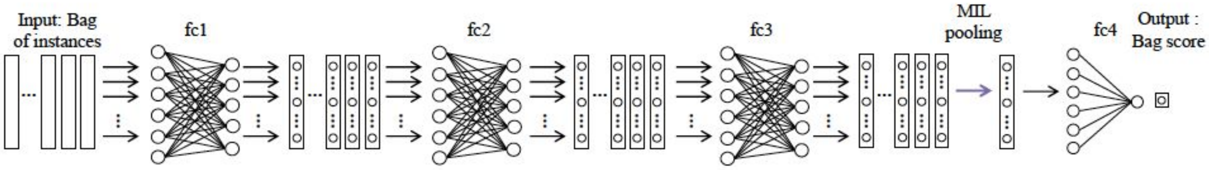


Figure 1: MI-Net - Input instances are aggregated into bag representation by first three fully connected layers and MIL pooling layer, and then use the last fully connected layer to predict bag probability.

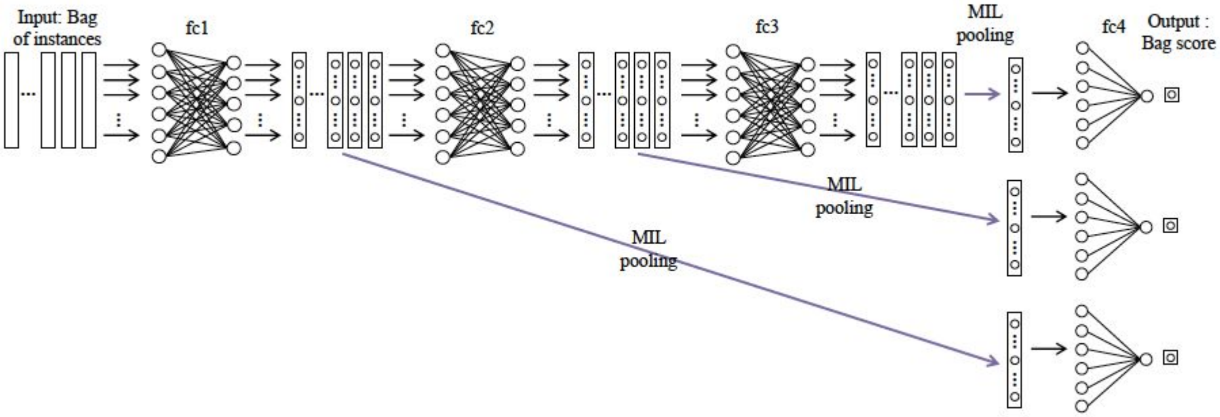


Figure 2: MI-Net with Deep Supervision - Each middle fully connected layer is followed by a MIL pooling layer and fully connected layer to compute bag scores. The loss function of MI-Net with Deep Supervision sums up all middle entropy losses to do backpropagation with SGD for training, and the average of each bag score is used for testing.

The default hyperparameters for both architectures are:

- Learning rate : 0.0005
- Pooling Layer : Max Pooling
- Weight Decay : 0.005
- Number of epochs : 5
- Momentum : 0.9

2.2 Classification of Sports Actions

Classification is a machine learning technique which is concerned of predictions of categorical labels. In this project the classes to be predicted are: action and non-action (background) which is a binary classification problem.

2.3 Evaluation Metrics

In a machine learning framework, the next step after performing a model training is to evaluate our model based on the metrics that we have defined. The context in which we are modeling is the classification of sports actions and hence, metrics are built on the comparison between true classes and the predicted classes. Some metrics that are common for classification problems are accuracy, precision, recall and F1-scores. In this project, F2-scores are also utilised to evaluate the performance. These metrics can be easily calculated based on the confusion matrix which defines the True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN):

- True positive (TP) is defined as the number of correctly predicted positive values.
- True negative (TN) is defined as the number of correctly predicted negative values.
- False positive (FP) is defined as the number of positively predicted values which are actually not positive. It is also called the Type 1 error.
- False negative (FN) is defined as the number of negatively predicted values which are actually not negative. It is also called the Type 2 error.

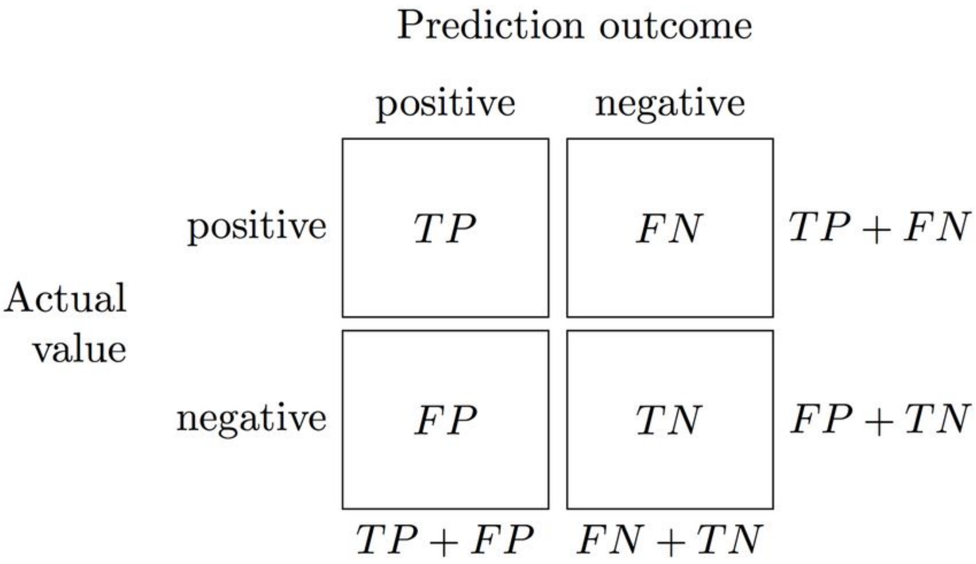


Figure 3: Confusion Matrix [4]

Based on these four values, we calculate the following evaluation metrics:

- **Accuracy** is ratio between the number of classes correctly predicted and the total number of classes to predict.

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

- **Precision** is the ratio of correctly predicted positive observations to the total predicted positive observations

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

- **Recall** is the ratio of correctly predicted positive observations to the all observation in actual class.

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

- The **F-beta score** is the weighted harmonic mean of precision and recall, reaching its optimal value at 1 and its worst value at

$$F_{\beta} - score = (1 + \beta^2) \frac{precision \times recall}{(\beta^2 \times precision) + recall}$$

In our project, we would like to maximize true positive and minimize false negatives. We would like to penalize a higher cost on False Negative (FN) than a False Positive (FP) and hence we utilise F2-score where beta is 2 on F_{β}-score defined above as the metric to evaluate the predictions.

3 Dataset

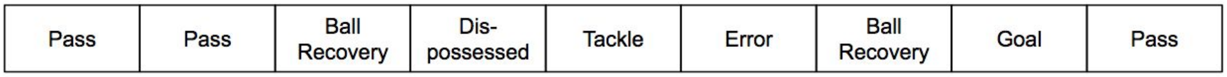
There are two main dataset that is being utilised in this project namely Opta data and the actual football videos from WildMoka. The sections below will elaborate on the datasets as well as the feature extraction and utilisation on these datasets.

In this project, each training batch is a bag. Instances definitions are described in each of the type of dataset in the sections below. A bag is labeled as positive, if there is at least one frame in the bag that is labelled as an action. A bag is labelled negative if all instances within the bag are background. We experimented with several ways of describing an instance.

3.1 Opta Data

Opta Sports is an international sports analytics company based in the United Kingdom. It provides data for 30 sports in 70 countries, with clients ranging from leagues to broadcasters and betting websites. In this project, Wildmoka provides us with the Opta labelled data with metadata features for the football videos that we are performing action detections on.

3.1.1 Feature Extraction



Pass	Pass	Ball Recovery	Dis-possessed	Tackle	Error	Ball Recovery	Goal	Pass

Figure 4: A sequence of events makes up a bag

The bags consists of instances which are events in opta data. Each instance is defined by its metadata. The metadata features that is utilised in this tasks are listed below.

Feature	Description
Type Array	Encoded array that signifies the type of action for an instance
Qualifiers Array	Encoded array that signifies the qualifiers for an instance
Outcome	Outcome of the action, as per listed in F24 Appendices
x	X position of the action
y	Y position of the action
Time passed	Time that has passed from the previous event to the current event

Table 1: Description of metadata features from Opta

The feature for each instance is generated by concatenating these metadata features. Each metadata feature is normalized with a min-max standardization except for the feature time passed. Min-max standardization is defined as:

$$x_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

There are two parameters to tune in order for us to generate the bags: sample size and stride. Sample size determines the number of instances within the bag. Stride determines the number of stride in a sequence of events in order to generate the next bag.

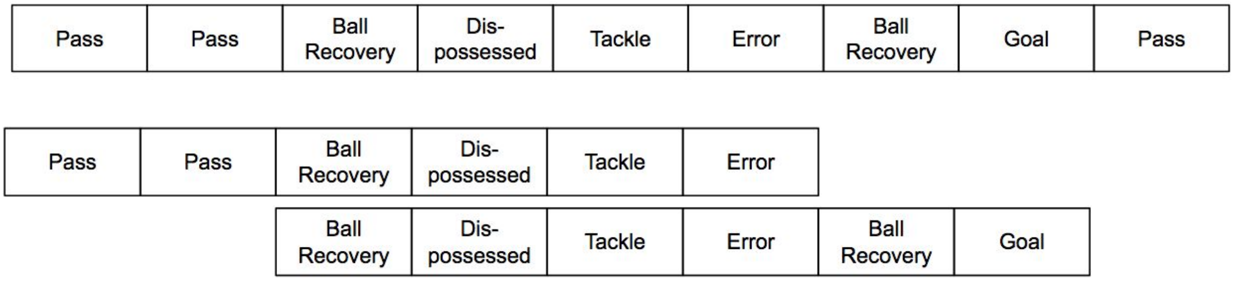

Pass | Pass | Ball Recovery | Dis-possessed | Tackle | Error | Ball Recovery | Goal | Pass
Pass | Pass | Ball Recovery | Dis-possessed | Tackle | Error
Ball Recovery | Dis-possessed | Tackle | Error | Ball Recovery | Goal

Figure 5: Original sequence of events and bags generated with sample size 6 and stride 2

3.1.2 Definition of action

The goal of the predictions in this task is to predict the snippets of the game defined in opta data that is of importance against the summaries generated by human as its gold dataset.

We define the positively predicted actions as *proposals* and the true label of the actions are based on the human generated summaries where if the action corresponds to the summary, it is a positive. We then obtain instances as defined in section 3.1.1 and a bag consists of these instances. Instances are sequential in order to preserve the order of the actions in the video. Unlike the traditional Multiple Instance Learning method where if a positive instance is in the bag, it is positive, we define the positive of the bag based on the percentage overlap of the bag against the true actions in summaries.

There are two ways proposed to evaluate the positivity:

$$(1)\quad \frac{number\ of\ overlap}{length\ of\ true\ proposals} > threshold$$

$$(2)\quad \frac{number\ of\ overlap}{min(length\ of\ bag,\ length\ of\ true\ proposals)} > threshold$$

The former (1) penalizes true proposals which are great in length i.e. long snippets of summaries in the true summaries and the latter (2) reduces this penalization.

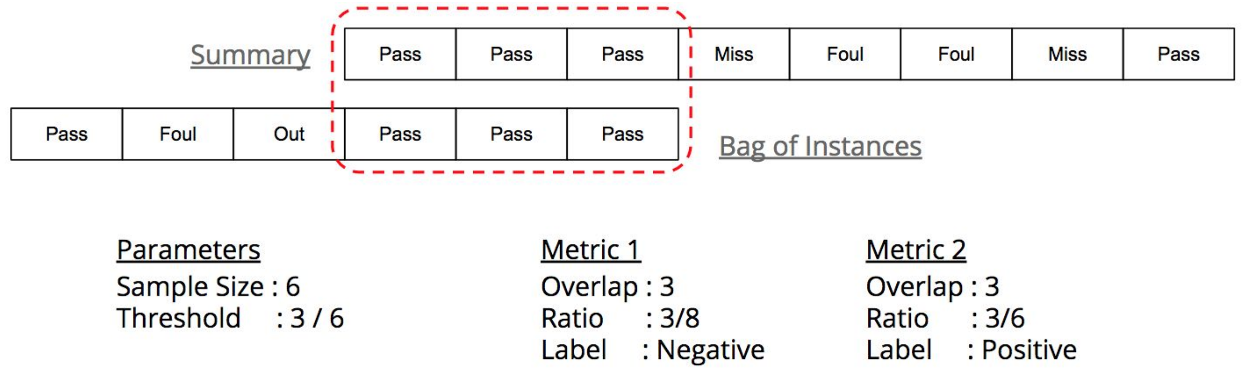


Figure 6: Metrics computation on long summaries

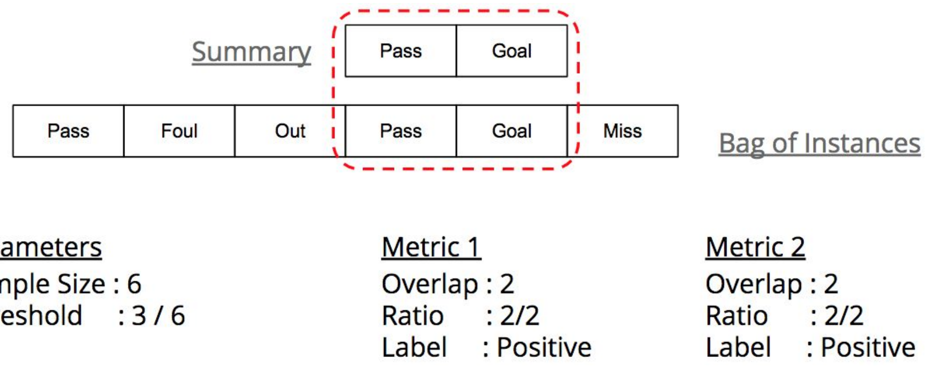


Figure 7: Metrics computation on short summaries

Figure 6 and Figure 7 above illustrates the impact of the two different metrics on the labels.

3.1.3 Label Evaluation

The methodologies utilised in this project produces bag-based predictions and in this tasks on predicting if the action will be in a summary, the evaluation metrics are based on instance based predictions. As such, post-processing on the bag predictions are applied in order to retrieve the instance predictions.

There are two ways in which the instance predictions are evaluated: (1) max operator on all bag predictions for the instance (2) log-sum-exp operator on all the bag predictions for the instance.

(1) $Probability = max\,(x_i)$

(2) $Probability = r^{-1}log(m^{-1} \sum_{i=1}^{n} exp\,(r \cdot x_i))$

where x_i is the probability of the instances in the bag, n is the number of bags with the instances and r is the smoothing parameter for the log-sum-exp operator where a greater r smooths it to a max operator and a smaller r smooths it to an mean operator.

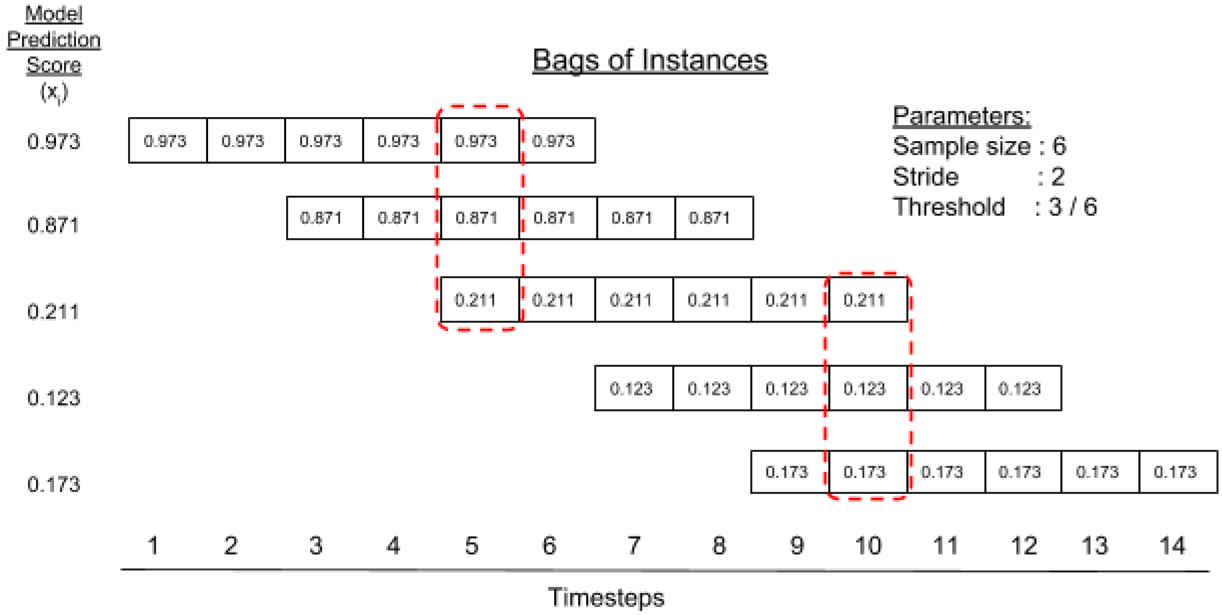


Figure 8: Computation of prediction values for instances. For an application of max operator on timestep 5, we obtain a value of 0.973

3.2 Video data

3.2.1 Feature Extraction

There are a total of 20 football videos that is being worked on in this task from the partner company of this project, WildMoka.

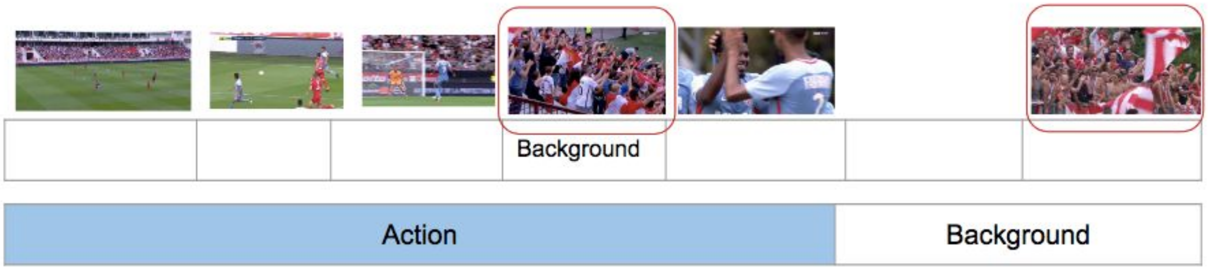


Figure 9: Illustration for the action and background events based on video data

Each instance in the bag is a frame from the video. The label of the instance is defined from its correspondence with the opta data which specifies an action or a background. In order to generate the features for the instance, the video is first split into frames with 25 frames per second (fps). We will then used the model of InceptionV3 pretrained on ImageNet to generate the features for each frame. The architecture for InceptionV3 is shown below.

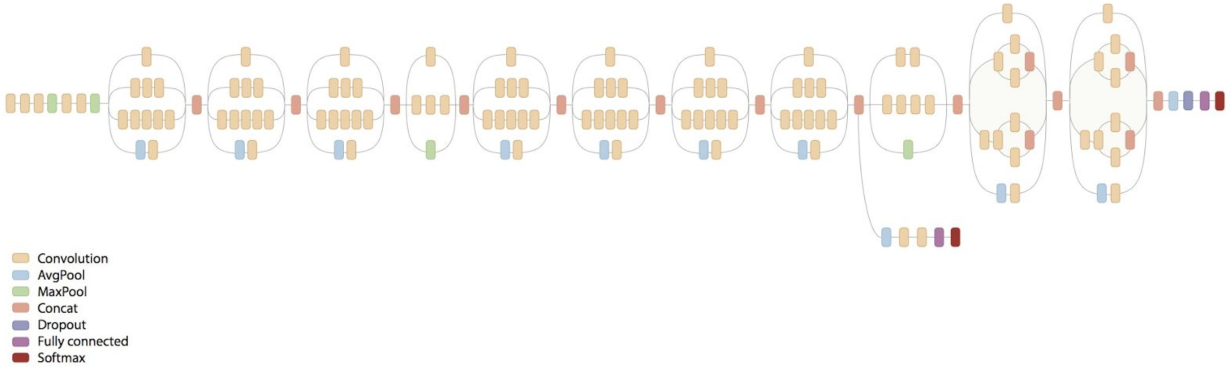


Figure 10: InceptionV3 Architecture

We obtain a feature of shape 8 x 8 x 2048 from the final layer of InceptionV3 as the representation of the frames and we flatten it to give us a feature vector of size 131072 as the input to our Multiple Instance Neural Network.

3.2.2 Label Evaluation

The evaluation for this dataset is determined by its bag prediction. The task optimizes for its bag accuracy. I.e. correctly classifying an action against a background.

4 Action Detection with Metadata Features

4.1 Experiment setup

The neural networks trained for each of the experiments with metadata features are setup with the architecture proposed by Wang et al, 2018 for both the MI-Net and MI-Net with Deep

Supervision architecture. Hyper parameter tuning at various levels such as feature generation, model hyperparameters and evaluation criterias are performed in order to find the model with the best metrics as well as deeper analysis in to the data that we are working with.

4.2 Results

The table below summarizes the results obtained from the different parameter tunings.

Positive Type	Architecture	Sample Size	Threshold	Model Acc	Instance Evaluation	Instance Metrics	Positivity Threshold
1	MI-Net	7	4/7	0.94193	Max	Mean Acc:0.641512596483 Mean F2:0.311693402934 Mean Precision:0.0901874875869 Mean Recall:0.852243676963	0.35
1	MI-Net	6	5/6	0.95504	Max	Mean Acc:0.699344675685 Mean F2:0.32485610721 Mean Precision:0.0994709563285 Mean Recall:0.790224404671	0.35
1	MI-Net	7	2/7	0.91837	Max	Mean Acc:0.633349548584 Mean F2:0.310915079052 Mean Precision:0.0895613534308 Mean Recall:0.86397861585	0.35
1	MI-Net with DS	7	2/7	0.91481	LSE	Mean Acc:0.597775400888 Mean F2:0.291938050551 Mean Precision:0.0816484798257 Mean Recall:0.866401595219	0.35
2	MI-Net	5	3/5	0.91403	Max	Mean Acc:0.785946258071 Mean F2:0.349961180911 Mean Precision:0.131106491297 Mean Recall:0.659499693166	0.5
2	MI-Net	5	3/5	0.91403	LSE	Mean Acc:0.783511414053 Mean F2:0.359735944873 Mean Precision:0.134926607231 Mean Recall:0.677682618256	0.35
2	MI-Net	10	7/10	0.87754	Max	Mean Acc:0.564537399517 Mean F2:0.295178601763 Mean Precision:0.0842322303876 Mean Recall:0.873039938964	0.35

From the results above, we observe that the definition of positivity of type 1 results in better model accuracy i.e. bag classification over the definition of positivity of type 2. However, the instance based accuracy and F2 scores are better than for the latter than the former. Evaluating the instances with LSE operator improves the recall and hence improving the F2 scores. The average size of ground truth are about 6 to 7 events in a sequence and hence we observe likewise that building bags of sample size 5 to 6 gives better performance over a bigger bag with sample size of 10.

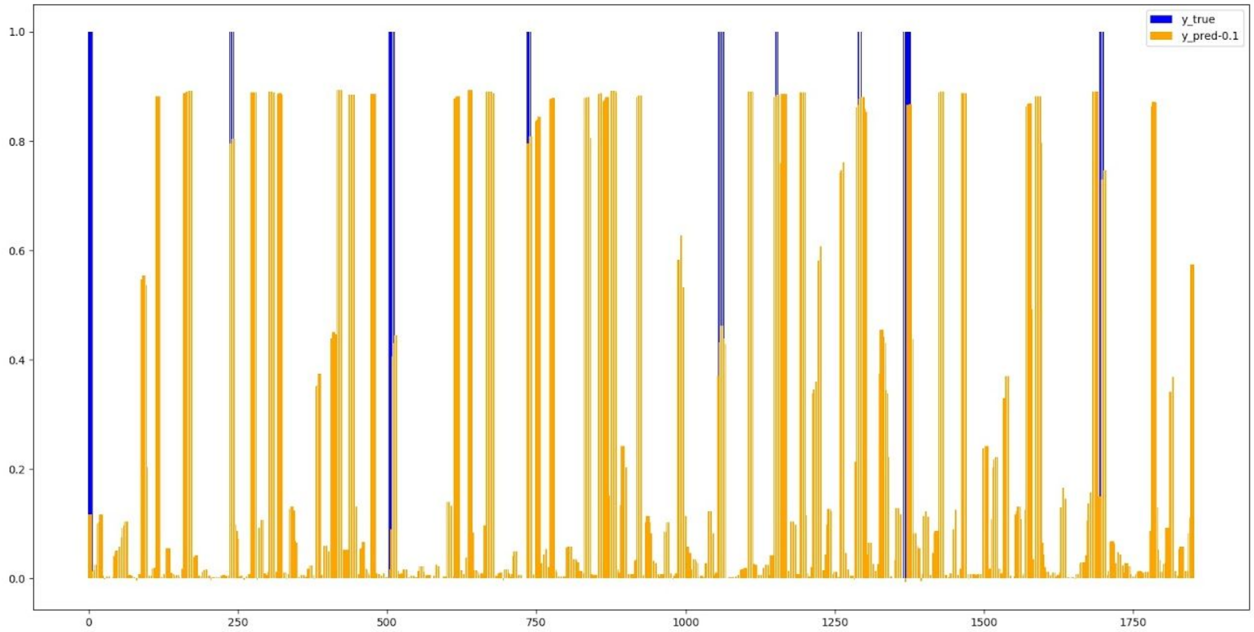


Figure 11: Plot of probability of being labeled as action over timesteps.

Positivity type (1) with instance evaluation with Max operator.

A visual analysis is performed and we observe that predictions of ground truth with bigger timesteps i.e. bigger area denoted in blue are not predicting as well and thus, the definition of positivity is being re-evaluated and the second definition of positivity is proposed as an option to potentially solve this problem.

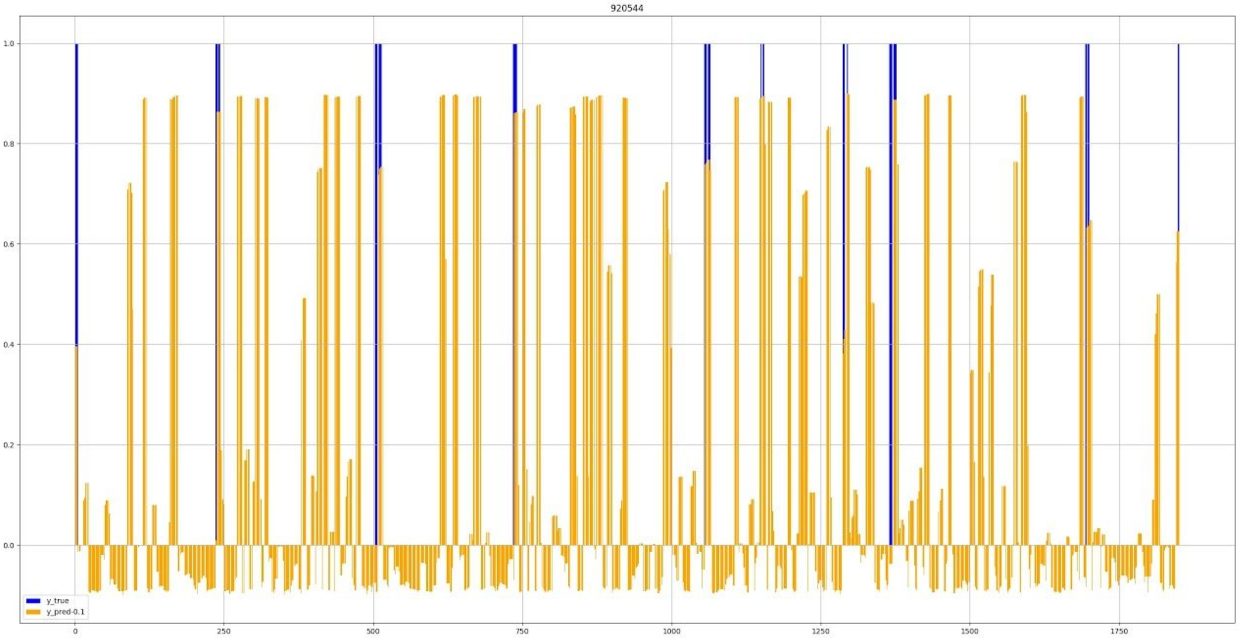


Figure 12: Plot of probability of being labeled as action over timesteps.

Positivity type (2) with instance evaluation with Max operator

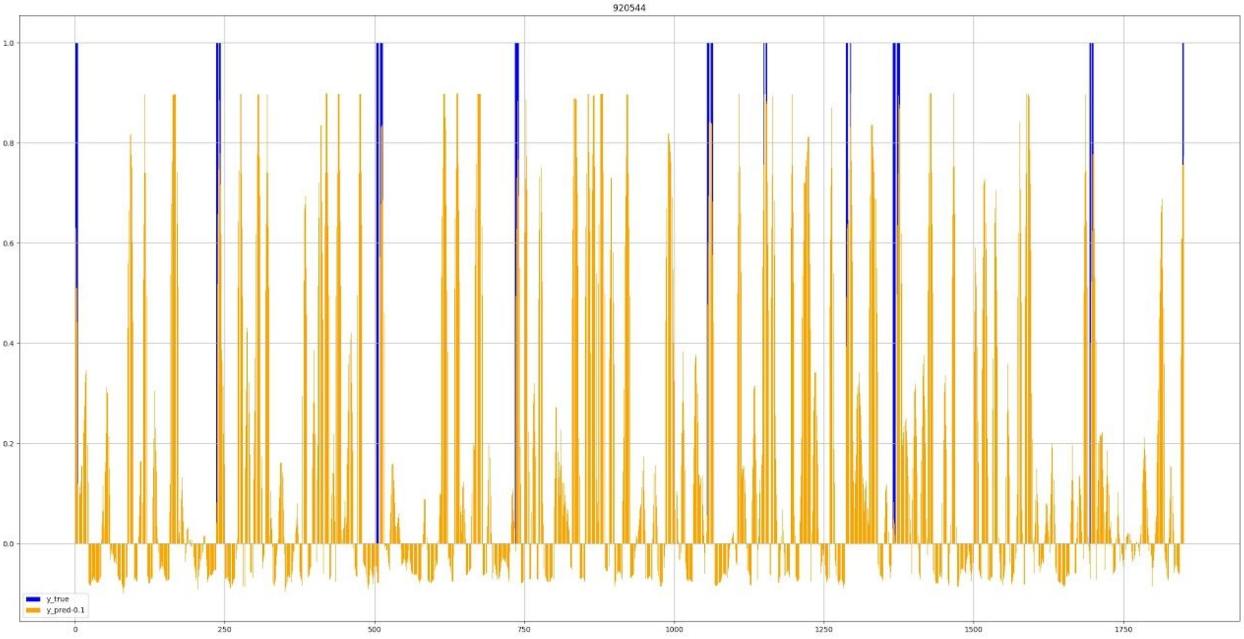


Figure 13: Plot of probability of being labeled as action over timesteps.

Positivity type (2) with instance evaluation with log-sum-exp operator

Observing both Figure 12 and 13 with type 2 definition of bag positivity, it can be observed that predictions for ground truth with bigger timesteps has shown more promising probability to be labelled as positive.

5 Action Detection with Video Features

5.1 Experiment setup

The neural networks trained for each of the experiments with metadata features are setup with the architecture proposed by Wang et al, 2018 for only for the MI-Net architecture. The architecture is trained with both AdamOptimizer as well as the originally proposed SGD optimizer. The original application of the MI-Net architectures proposed by Wang et al, are trained on datasets which are of much smaller size than the data that is being generated on video and hence the model fitting is done with a generator and hyperparameter tuning are being conducted in order to find the best application of the model to the dataset.

Due to the size of the video features that is being generated, we observe a limitation in the computing power to load all the features in memory for training or processing. During training, the values in the hidden layers deviates from zero mean and unit variance and hence changing the distribution of the input to the next layer. These changes affects the learning rate of the networks and hence deviating from the original architecture proposed, in some experiments, Batch Normalization layers are added to each fully connected layer. Note that the Batch Normalization are applied on each batch which represents only one bag of instances.

5.2 Results

The table below summarizes the results of the model trained with Multiple Instance Learning.

Model	Optimizer	Dropout	Model Accuracy
MI-Net	SGD	0.5	Not learning
MI-Net	SGD	None	0.71551
MI-Net	Adam	None	0.52667
MI-Net	SGD	0.1	0.68424

Training the neural network with the original architecture proposed in has shown results that the model is not learning over the epochs. To find the root to this problem, we experimented with the learning rate and added Batch Normalization layers. Despite that, it does not show any learning ability. On removal of the dropout, we observe that the network is learning and hence, the original architecture is underfitting on the video features. Experiments with Adam Optimizer over the Stochastic Gradient Descent (SGD) optimizer has shown that SGD has given better results.

6 Conclusion and future work

The work that has been done in this project is part of the bigger architecture in the context of sports video summarization as part of the paper submission for ACM Multimedia Conference 2019. Modelling with Opta data has shown optimistic results and the preliminary results on modelling with video features requires further optimization. The goal of the future work is to retrieve more substantial information from video features to achieve better results from the current results we observe from utilising the metadata features from Opta.

Bibliography

[1] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, “Solving the multiple instance problem with axis-parallel rectangles,” Artificial Intelligence, vol. 89, no. 1, pp. 31–71, 1997.

[2] J. Amores, “Multiple instance classification: Review, taxonomy and comparative study,” Artificial Intelligence, vol. 201, pp. 81–105, 2013.

[3] Xinggang Wang, Yongluan Yan, Peng Tang, Xiang Bai, and Wenyu Liu. 2018. Revisiting multiple instance neural networks. Pattern Recogn. 74, C (February 2018), 15-24. DOI: https://doi.org/10.1016/j.patcog.2017.08.026

[4] Masías, Víctor Hugo & Valle, Mauricio & Morselli, Carlo & Crespo, Fernando & Vargas Schüler, Augusto & Laengle, Sigifredo. (2016). Modeling Verdict Outcomes Using Social Network Measures: The Watergate and Caviar Network Cases. PloS one. 11. e0147248. 10.1371/journal.pone.0147248.