Project Report : CS 7643

Adithya R Embar, John S Gilmore, Payaam Emami
Georgia Institute of Technology
Atlanta, GA 30332, USA
{aembar3, jgilmore39, pemami7}@gatech.edu

Abstract

This project focuses on improving performance in the SoccerNet Action Spotting Challenge by iterating on the benchmark model presented by Deliège et al. (2020). The benchmark, a CNN-based architecture featuring a temporal pyramid for feature extraction and a segmentation and spotting module, achieved a test mean average precision (mAP) of 41.93%. Our approach sought to enhance the model's contextual understanding over the temporal dimension, hypothesizing that attention mechanisms could address the challenge effectively. We explored several model variations, including the application of multi-head attention and the integration of squeeze-and-excitation (SE) blocks after the temporal pyramid. Our experimental results demonstrated improved performance, with the Multi-head Attention model achieving the highest test mAP of 43.34%. These results underscore the value of incorporating attention-based mechanisms for temporal action spotting in soccer videos, building on the foundational work of Deliège et al.

1. Introduction/Background/Motivation

This project aims to solve the problem of automating the process of detecting action events in soccer footage. The goal of this project is to improve the performance of a benchmark convolutional neural network, provided by the SoccerNet research community, by introducing attention mechanisms into this baseline model. The three mechanisms that were implemented are: self-attention, multi-head attention, and squeeze-and-excitation blocks.

The SoccerNet research community provides challenges as well as a large-scale dataset for soccer video understanding. [3] The SoccerNet dataset was created to address the task of action spotting, which is also the selected challenge for this project. The SoccerNet action spotting challenge is formed as a multiclass classification problem, where the labels (i.e. actions) are: Penalty, Kick-off, Goal, Substitution, Offside, Shots on target, Shots off target, Clearance, Ball out of play, Throw-in, Foul, Indirect free-kick, Direct free-kick, Corner, Yellow card, Red card, Yellow and red card (together). Event annotations are obtained from publicly available match reports on league websites, which provide a summary of key actions and their respective game minutes. For instance, a "goal" event is annotated at the precise moment the ball crosses the goal line. Similar granular definitions are used for "cards" (the moment the referee issues the card) and "substitutions" (the moment a player steps onto the field). This process ensures consistency and high-quality temporal accuracy. [2]

The dataset consists of 500 soccer game broadcast videos from six main European leagues, covering three seasons from 2014 to 2017 and a total duration of 764 hours. The SoccerNet development kit also provides extracted features at 2 frames per second by a Resnet Model. The challenge set is composed of 50 separate games. [3] [2]

Today, human annotators manually perform action spotting in soccer footage, a time-consuming and non-scalable process. As the soccer footage data increases, the number of annotators or the time spent by annotators will also increase. However, there are a number of services that provide AI solutions for action spotting. For example, there are services that provide automated sports highlight creation. Automating highlight creation is a practical use case for action spotting neural networks. However, it is a niche problem.

More importantly, accurately and automatically spotting actions lay the foundation on which future work can be built. For example, future systems can leverage neural networks that accurately perform action spotting to provide sports analysis in real time.

1.1. Benchmark model background

The Context-Aware Model (CAM) serves as our chosen benchmark architecture for the SoccerNet action spotting challenge. The architecture leverages both temporal and contextual features to produce segmentation and action spotting predictions. It receives data in the format of (batch, chunk size, features) where chunk size is a preselected number of frames. The model outputs a segmentation matrix that provides a score for each class for every frame,

and a spotting matrix that identifies specific time frames for key actions with refined confidence and class predictions. The segmentation matrix is used as an input for the spotting module.

The architecture begins with two convolutional layers, which reduce the feature dimensionality and extract low-level temporal features. The first convolution maps the input features into a higher-dimensional space. The second convolution further processes this data, refining the feature representation.

A hierarchical Temporal Pyramidal Module (TPM) is employed to capture multi-scale temporal patterns. This module comprises four parallel convolutional branches, each with different receptive field sizes, allowing the model to process varying temporal granularities. The outputs of these branches, along with the base convolutional features, are concatenated to form a representation of the temporal context.

The segmentation task involves predicting the probability of each action class at every frame. A convolutional layer processes the concatenated features from the TPM, followed by reshaping and normalization through batch normalization. The final segmentation scores are computed using capsule-based feature norms, emphasizing the magnitude of class-relevant features.

The action spotting branch uses the segmentation scores as input, combining them with the capsule representations to enhance detection. A series of convolutional and pooling layers progressively reduce the temporal resolution, allowing the model to focus on key temporal regions. The module outputs confidence scores which predict the likelihood of a detected action, and class scores which assigning a class label to each detected action.

These outputs are concatenated to provide a comprehensive detection result for each identified event.

The model also makes use of a custom loss function which is designed to penalize the frames far-distant from the action and steadily decrease the penalty for the frames gradually closer to the action. The frames just before the action are not penalized to avoid providing misleading information as its occurrence is uncertain. However, those just after the action are heavily penalized as we know that the action has occurred. The Adam optimizer is used.

2. Approach

Our primary avenue of improvement was targeting the feature extractor and temporal pyramid in the benchmark for improvement. This component relies on hard-coded range selections within the receptive field to provide the downstream modules with adequate information on the source data across a few temporal windows (immediate, short, medium, long). We thought that while this may provide some domain expertise to guide early training, allowing the model to utilize attention to capture inter-temporal relationship via training will make a stronger model. One could imagine a situation where there is a foul followed by a red card, however in between the foul and red card some crowd footage is included. Our goal with applying attention based feature extraction was to be able to better capture a connection between the foul and the subsequent red card action without wasting attention on the likely useless crowd footage in between. To do this we started by replacing the initial CNN based feature extractor and temporal pyramid with a transformer.

For reasons expanded on in Section 3, we elected to restore the TPM but keep developing the idea of incorporating attention for more effective inter-temporal relationship identification. This led us to process the output of the TPM first with a multi-head attention layer and then with a squeeze and excitation based approach. Additionally we made changes to improve the training process through introduction/addition of dropout and batch normalization, as well as the eventual addition of temporal smoothing in the segmentation module. Our contributions are new to the model but not necessarily new to the action spotting problem with there being abundant existing work on the effectiveness of attention/transformers when applied to computer vision and particularly video processing.

We conducted a series of experiments to evaluate how effectively different model configurations captured temporal and contextual relationships. This involved analyzing saliency, average class probabilities, normalized confidence scores, and average activations to evaluate class variability, temporal focus, and feature prioritization across the models. These experiments provided insights into the strengths and limitations of each architecture.

We anticipated a few problems. First our dataset is very large with the raw video data being several terabytes. Given the time constraints of the project we feared this would make our training process too slow in aggregate and limit effective experimentation. As a result we elected to use the ResNet extracted features provided by the challenge as the input to our model (this is also what our benchmark comparison model did so this kept consistency for comparison). We did have fear about the effectiveness of applying transformers/attention to features already extracted by a ResNet but research from Srinivas et al [6] and Wu et al [8] provided evidence that processing ResNet features with a transformer/self attention heads has potential for improved performance.

The first thing we tried was not successful. Our transformer based feature extractor results in a significant reduction in mAP which forced us to consider the value of the TPM and develop alternative approaches which led to the subsequent multihead attention and squeeze and excitation approaches. Other problems encountered included

our models overfitting as well as training instability when adding complexity

We utilized one branch of the benchmark code provided by the SoccerNet repo as a starting point. [3] The work there is associated with the paper "A Context-Aware Loss Function for Action Spotting in Soccer Videos." [1]

3. Experiments and Results

The average-mAP (mean Average Precision) metric, as defined in SoccerNet: A Scalable Dataset for Action Spotting in Soccer Videos (Giancola et al.) [2], is used to evaluate the models and measure success. This metric is commonly used in object detection, action detection, or event spotting tasks to evaluate the performance of models when precise temporal or spatial localization is required. [3]

Average Precision (AP) measures the quality of predictions by combining precision (the fraction of true positives among all predicted positives) and recall (the fraction of true positives among all actual positives). The mean Average Precision (mAP) average of the AP values over all classes or events.

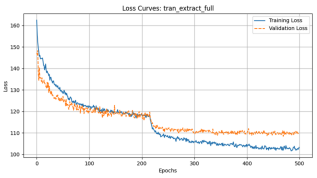
We used the PyTorch deep learning framework to complete our work.

Model	Average m-AP
Baseline Context Aware	0.4193
Transformer Extractor	0.2637
Multihead Attention	0.4334
Squeeze-and-Excitation Block	0.4300

Table 1: Comparison of models using average m-AP metric

3.1. Model Development

(a) Overfitting in benchmark model



(b) Transformer loss curve

One important finding from our initial analysis of the benchmark model is that the model notably overfits the training data with validation loss generally ranging from being 20% to 50+% greater than training loss1a.

Per section 2, we began development with replacing the starting convolution layers and the TPM with a transformer encoder. We will refer to the model developed as the transformer feature extractor model (TFEM). The encoder is a two-layer Transformer Encoder, employing 8 attention heads and a feedforward dimension of 2048. The transformer extractor based model achieved a map of 26.37%, which fell short of the baseline model's performance. We did note though that our transformer based feature extractor did overfit notably less than the benchmark model1b.

We hypothesize that the decline in mAP is likely due to underestimating effectiveness of the pre-selected pyramid windows for capturing informative features based on domain expertise. It is also possible through the transformer, though we should be better capturing inter-temporal relationships, we are failing to capture the low level features that CNNs excel at capturing via the inductive bias of their kernels. It is possible that the effectiveness of the preselected windows actually leads to the overfitting we are seeing but also still provides a higher ceiling for mAP performance.

Building upon insights gained from the failure of the TFEM, we aimed to combine the strengths of the original temporal pyramid with the adaptability of attention-based methods. Thus, our revised approach maintained the original pyramidal structure—ensuring that multi-scale temporal patterns remain explicitly encoded—while introducing a transformer module downstream to integrate these representations in a more contextually aware manner. We will refer to the model developed in this portion as the multihead attention model (MHAM).

Specifically, we preserved the initial convolutional layers and temporal pyramid module as-is, generating a multi-scale feature map. We then fed the concatenated pyramid outputs through a lightweight transformer encoder with positional encodings. This choice was motivated by the desire to restore the TPM but still capture the inter-temporal explainability of self attention. We saw this as particularly valuable because at the end of the TPM the different windows are concatenated together. Applying self attention to this concatenated matrix should help us capture meaningful relationships between those windows that the purely CNN based approach is not offering. The multi-head self-attention mechanism allowed the model to re-weight and refine the multi-scale features, focusing on temporal relationships that might not be equally salient at all levels of the pyramid.

Our initial run of this architecture features a self attention feed forward dimension of 512 and 4 heads; it achieved a mean Average Precision (mAP) of 42.36%, surpassing the original benchmark and demonstrating that the hybrid approach better aligned with the nature of the underlying data. The improved performance can be attributed to the synergy between the explicit multi-scale representations from the temporal pyramid and the transformer's ability to dynamically highlight relevant temporal segments. In essence, the transformer no longer had to "discover" scales of relevance from scratch—these were already provided—allowing at-

tention mechanisms to focus on integrating and contextualizing these features, thereby yielding more accurate and robust action spotting predictions.

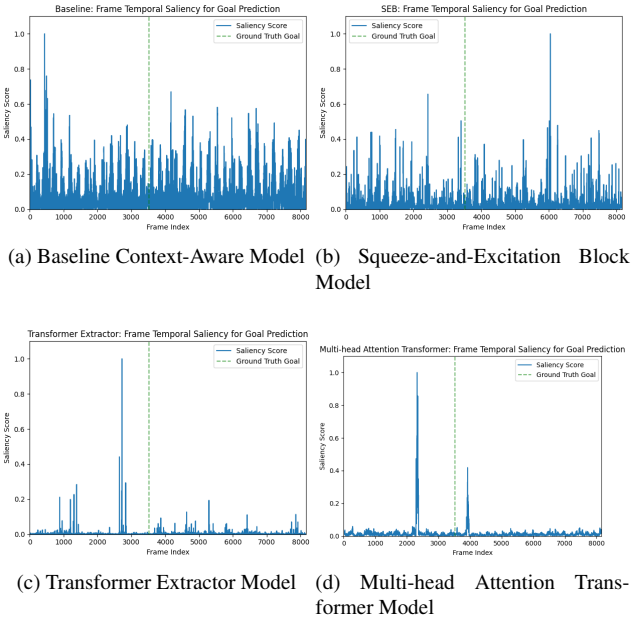
Before refining the MHAM, we wanted to try one more major architecture change. We developed a variant of the model incorporating Squeeze-and-Excitation (SE) blocks into the temporal pyramid, accordingly the model developed in this portion will be referred to as the squeeze and excitation model (SEM). These blocks augment the architecture's ability to focus on the most informative channels of the extracted features. By applying a global average pooling operation followed by a small fully connected network and a sigmoid nonlinearity, each SE block learns to reweight feature maps adaptively. This acts somewhat similarly to attention mechanisms: just as attention can highlight crucial temporal regions, the SE blocks emphasize or suppress particular feature channels, guiding the model to attend to the most relevant aspects of the representation.

To implement this, we retained the original temporal pyramid structure to capture multi-scale temporal patterns, but applied grouped convolutions within the pyramid layers and integrated SE blocks on each branch. These modifications were grounded in similar ideas to the MHAM approach being that the strong priors of the TPM could benefit from refined focus, which channel-wise attention mechanisms are well-suited to provide. The grouped convolutions encourage specialization within subsets of the feature channels, and SE blocks then selectively amplify or diminish their contributions. This process yields a discriminative feature set entering the subsequent segmentation and detection modules.

Empirically, these modifications increased the model's mAP to 43.00% (not rounded). The enhanced performance can be understood as a direct result of enabling the network to prioritize informative channels that contribute to distinguishing actions. By combining the strengths of multi-scale feature extraction and fine-grained channel re-weighting, the architecture better captures the subtle temporal and contextual cues necessary for accurate action spotting.

We still noted overfitting in both the SEM and MHAM as well as some training instability in the MHAM when feedforward dimension size is increased. To attack this we added additional regularization through the addition of new batchnorm layers at the conclusion of the extraction and segmentation modules as well dropout following the extraction module. We added weight decay to the optimizer and thus changed from the original Adam optimizer to the AdamW optimizer for it's superior handling of weight decay[5]. Additionally, we added a temporal smoothing layer to the end of the segmentation module. The temporal smoothing encourages smoother, more coherent predictions over time, but also leads to more stable training. The refinement step enforces temporal consistency, making the model's predictions less erratic and improving gradient stability during backpropagation. This addition did not move the needle for the SEBM but it unlocked greater performance in the MHAM (in combination with hyperparameter tuning, see Table 3 in appendix, changed to 8 heads and 1024 feed forward dim)3 and led to our best mAP score of 43.34%.

3.2. Temporal Saliency Visualizations for Goal Prediction



(a) Baseline Context-Aware Model (b) Squeeze-and-Excitation Block Model

(c) Transformer Extractor Model (d) Multi-head Attention Transformer Model

Figure 1: Temporal Saliency Visualizations for Goal Prediction using different models.

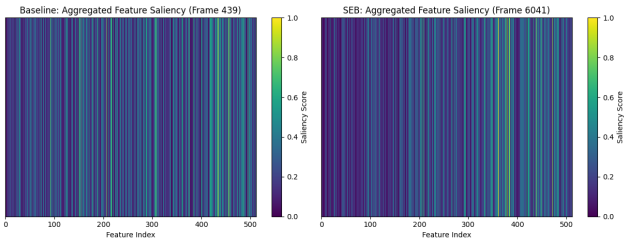
In figure 1, the temporal saliency plots showcase different patterns in how each model focuses on video frames.

The baseline context-aware model (figure 1a), shows a scattered saliency distribution with numerous frames that have high saliency scores, which suggests uncertainty about which frames influence its predictions. The squeeze-and-excitation block (SEB) model (figure 1b), somewhat improves on this, as we can see certain frames stand out more. However, the saliency distribution is still scattered, similar to the baseline model. The SEB model shows slight improvements with less noise, however, it doesn't drastically impact the model's temporal focus. More importantly, in figures and 1b, we can see that the frames with highest saliency scores are not near the ground truth goal frame. This suggests that these two models are not able to effectively leverage temporal context to predict the goal frame.

On the other hand, the transformer-based models showcase more selective attention. The transformer extractor (TE) model (figure 1c) identifies a select few key frames

with distinct saliency scores that are near the ground truth goal frame. The multi-head attention transformer model (figure 1d) shows better results by having even fewer frames with high saliency scores, that are also near the ground truth goal frame, showcasing that these model can identify specific moments (i.e. frames) that are critical to their goal predictions. This also suggests they excel at using temporal context to identify the most relevant moments. From these plots, we can gather that the CNN-based models distribute their attention broadly, while the transformer-based models use attention to hone in on specific and fewer frames, suggesting better temporal understanding.

3.3. Feature Saliency Visualizations for Goal Prediction



(a) Baseline Context-Aware Model (b) Squeeze-and-Excitation Block Model

(c) Transformer Extractor Model (d) Multi-head Attention Transformer Model

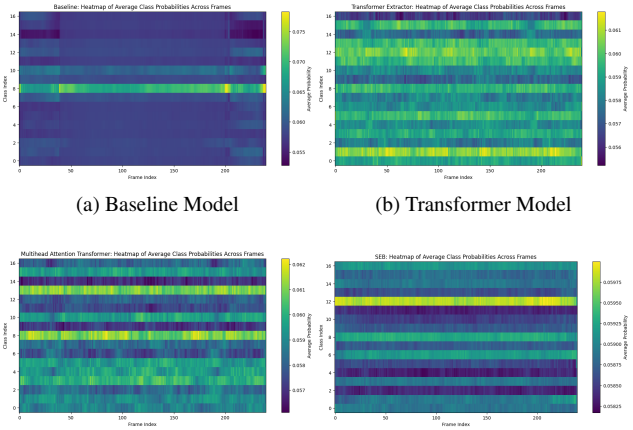
Figure 2: Aggregated Feature Saliency Visualizations for Goal Prediction using different models.

In figure 2, the feature saliency plots showcase how the models value individual feature dimensions at a given frame. The frames chosen for each model are the ones with the highest saliency scores from the temporal saliency plots (figure 1).

Each model shows different feature importance across the feature dimensions. Each vertical line in these plots corresponds to a feature derived from the pretrained ResNet features. No specific dimension consistently stands out nor can we see any obvious pattern. This suggests that no single feature dominates the goal predictions for any model.

When comparing temporal saliency with feature saliency, we observe that the main difference between the models is not which features they prioritize, but how they interpret these features over time (i.e. frames). The models do not contain significant similarities for specific features that are weighted more heavily towards goal prediction, shown in figure 2. However, the transformer-based models appear to be superior due to their ability to focus attention on specific frames over time (shown in figure 1).

3.4. Average Class Probability Visualizations



(a) Baseline Model (b) Transformer Model

(c) Multi Head Attention Transformer Model (d) SEB Model

Figure 3: Heatmaps of Average Class Probabilities Across Frames

In figure 3 the Baseline Model reveals an emphasis on a single class 8. This indicates that the model focuses heavily on one type of action while not focusing on others, leading to a lack of balanced representation. This behavior likely stems from the hierarchical temporal pyramidal module, which, although captures multi-scale patterns, may fail to adequately account for nuanced temporal relationships[2].

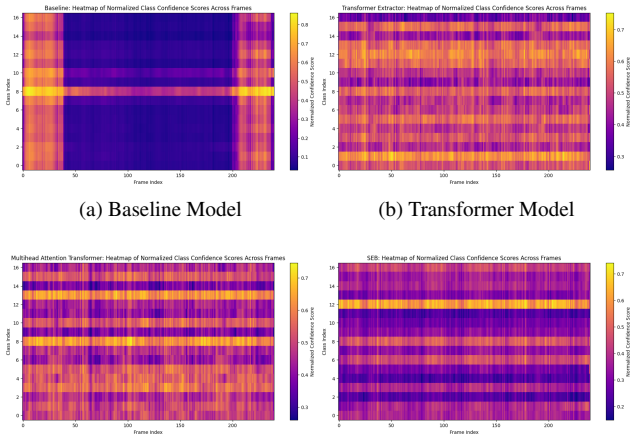
The Transformer model's heatmap shows a more uniform distribution of probabilities across all classes. Despite its broader focus, the lack of distinct activations for any specific class suggests difficulty in modeling key temporal dependencies which are required for effective action spotting[7]. This is evident from its low mAP of 0.2637, showing that its broader focus reduces effectiveness in event differentiation.

The Multi-head Attention heatmap demonstrates higher variability across multiple classes compared to the previous models. This improved differentiation across classes highlights the model's ability to capture contextual cues and temporal dependencies[7]. This finding aligns with its highest mAP of 0.4334, and further shows its high ability in contextual learning and action-spotting capabilities.

The SE Block heatmap shows us a fairly balanced distribution of probabilities, with class 8 showing notable activity, between 0.064 and 0.067. This tells us that the SE

blocks enhance the model's focus on relevant temporal features while maintaining some diversity in activation across classes[4]. That being said, the lack of sharp peaks as seen in the Multi-head Attention Model, indicates a reduced ability to fully capture the variance of the soccer actions.

3.5. Class Confidence Scores Visualizations



(a) Baseline Model (b) Transformer Model

(c) Multi Head Attention Transformer Model (d) SEB Model

Figure 4: Heatmaps of Normalized Class Confidence Scores Across Frames

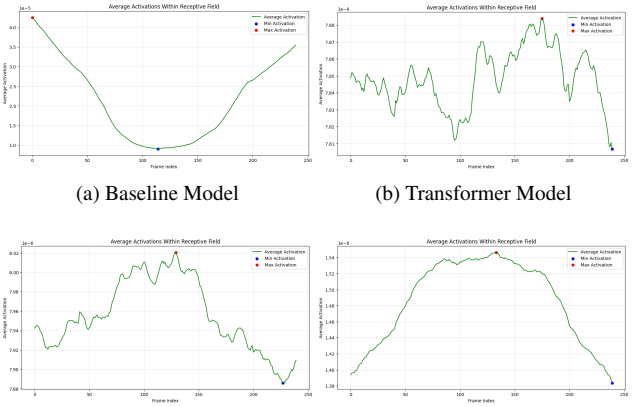
In figure 4 the heatmap for the Baseline Model shows a concentrated confidence in one particular class 8. This reflects the model's tendency to heavily emphasize one specific class[2]. This pattern indicates that the Baseline Model's architecture seems to overfit on a dominant class, resulting in limited generalization[2].

The heatmap for the Transformer Extractor displays a more even and uniform distribution of confidence scores across all classes[7]. The lack of pronounced confidence peaks in any particular class suggests that the model struggles to establish a strong distinction between specific types of soccer events[7].

The Multi-head Attention Model's heatmap shows higher confidence scores across multiple classes. The higher normalized confidence scores suggests that the Multi-head Attention mechanism improves the model's ability to isolate and emphasize relevant features[7]. Additionally, the higher variability in the heatmap across frames suggests a stronger capability for temporal modeling[7].

The SE Block-enhanced model exhibits a fairly balanced confidence pattern. While its distribution is similar to the Multi-head Attention model, the confidence scores are lower on average. The SE Blocks likely enhance feature selection but do not leverage temporal attention, which could limit how they capture subtle temporal dependencies[4].

3.6. Average Activations Visualizations



(a) Baseline Model (b) Transformer Model

(c) Multi Head Attention Transformer Model (d) SEB Model

Figure 5: Average Activations Within Receptive Field

In figure 5, the baseline model exhibits a significant range in activation values, where we see this pattern shows a downward trend initially, followed by a gradual increase towards the end. The high activation at Frame 0 suggests a strong response to initial features; however, the decline and mid-frame low values highlight the model's inability to maintain strong activations[2].

The transformer model's activations have bigger peaks, with a narrower range between the minimum and maxiumum activations. The fluctuations suggest a lack of focus on key frames, while peaks around Frame 175 hint at a mid-sequence response, but the flat trend indicates ineffective use of temporal variations[7, 2].

The multi-head attention model shows a slightly broader range in activations, with the activation curve showing distinct fluctuations, suggesting that the multi-head attention mechanism effectively focuses on temporal features[7]. The variability and the ability to emphasize specific frames align with the model's highest mAP score[7].

The SEB model features the lowest overall activations, with a fairly symmetric trend, peaking around the mid-frame and declining steadily toward the end. Despite lower activation values, the SEB model's mid-sequence performance shows its feature prioritization makes it reliable[4].

4. Conclusion

In conclusion, our work demonstrates the potential of attention-based architectures for temporal action spotting, delivering results beyond the baseline CAM model. We also highlight the challenges of integrating attention mechanisms and emphasize the need for thoughtful design to prevent overfitting. These insights provide a strong foundation for future exploration in deep learning to improve the automation of event detection in soccer videos.

5. Work Division

Table 2 showcases the delegation of work among our team members.

6. Appendix

Table 3 Showcases recorded test mAP for different multihead attention hyperparameter combinations when trained on a sub-sample of 50 games for 50 epochs.

References

[1] Adrien Deliège Deliège, Anthony Cioppa, Silvio Giancola, Meisam J. Seikavandi, Jacob V. Dueholm, Kamal Nasrollahi, Bernard Ghanem, Thomas B. Moeslund, and Marc Van Droogenbroeck. Soccernet: A scalable dataset for action spotting in soccer videos. *arXiv preprint arXiv:1912.01326*, 2020. 3

[2] Silvio Giancola, Mohieddine Amine, Tarek Dghaily, and Bernard Ghanem. Soccernet: A scalable dataset for action spotting in soccer videos. *arXiv preprint arXiv:1806.07737*, 2018. 1, 3, 5, 6

[3] Silvio Giancola, Adrien Deliège, Domenico Napolitano, Tarek Dghaily, and Bernard Ghanem. Soccernet: Action spotting and dataset for holistic understanding of broadcast soccer videos. https://www.soccer-net.org/tasks/action-spotting, 2024. Accessed: 2024-12-07. 1, 3

[4] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu. Squeeze-and-excitation networks. *arXiv preprint arXiv:1709.01507*, 2019. Accessed: 2024-12-07. 6

[5] lya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2019. Accessed: 2024-12-06. 4

[6] Aravind Srinivas, Tsung-Yi Lin, Niki Parmar, Jonathon Shlens, Pieter Abbeel, and Ashish Vaswani. Bottleneck transformers for visual recognition. *arXiv preprint arXiv:2101.11605*, 2021. Accessed: 2024-11-10. 2

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2023. Accessed: 2024-12-07. 5, 6

[8] Bichen Wu, Chenfeng Xu, Xiaoliang Dai, Alvin Wan, Peizhao Zhang, Zhicheng Yan, Masayoshi, Tomizuka, Josepj Gonzalez, Kurt Keutzer, and Peter Vajda. Visual transformers: Token-based image representation and processing for computer vision. *arXiv preprint arXiv:2006.03677*, 2020. Accessed: 2024-11-10. 2

Student Name	Contributed Aspects	Details
Adithya R Embar	Visualizations, Analysis, Conclusion	Analyzed average activation score and performed heatmap analysis on all four models for average class probability and class confidence for all 17 classes.
John S Gilmore	All new models	Designed, developed, and tested the transformer extractor, multihead attention, and squeeze and excitation models. Performed mAP based analysis on said models.
Payaam Emami	Visualizations and analysis	Leveraged the model checkpoints, provided by John, to generate the temporal and feature saliency visualizations for goal prediction and provided analysis between the models.

Table 2: Contributions of team members.

Head Count	Feedforward Dimension Size	Test mAP
2	256	21.26%
2	512	19.86%
2	1024	21.23%
4	256	21.03%
4	512	12.93%
4	1024	21.72%
8	256	14.59%
8	512	20.42%
8	1024	22.14%

Table 3: Showcases recorded test mAP for different multihead attention hyperparameter combinations when trained on a sub-sample of 50 games for 50 epochs.