S-amba: A Multi-View Foul Recognition in Soccer Through a Mamba-Based Approach

Henry O. Velesaca[1,2][a], Alice Gomez-Cantos[1,2][b], Abel Reyes-Angulo[3][c] and Steven Araujo[1][d]
[1]*ESPOL Polytechnic University, FIEC, CIDIS, Campus Gustavo Galindo, Guayaquil, Ecuador*
[2]*Software Engineering Department, Research Center for Information and Communication Technologies (CITIC-UGR), University of Granada, 18071, Granada, Spain*
[3]*Michigan Technological University, Houghton MI, U.S.A.*

Keywords: Multi-View Foul Recognition, Mamba, Computer Vision.

Abstract: In this work, we propose a novel Mamba-based multi-task framework for multi-view foul recognition. Our approach leverages the Mamba architecture's efficient long-range dependency modeling to process synchronized multi-view video inputs, enabling robust foul detection and classification in soccer matches. By integrating spatial-temporal feature extraction with a multi-task learning strategy, our model simultaneously predicts foul occurrences, identifies foul types, and localizes key events across multiple camera angles. We employ a hybrid loss function to balance classification and localization objectives, enhancing performance on diverse foul scenarios. Extensive experiments on the SoccerNet-MVFoul dataset demonstrate our method's superior accuracy and efficiency compared to traditional CNN and Transformer-based models. Our framework achieves competitive results, offering a scalable and real-time solution for automated foul recognition, advancing the application of computer vision in sports analytics. The codebase is publicly available at https://github.com/areyesan/Mamba-Based_MVFR for reproducibility.

1 INTRODUCTION

Automated foul recognition in soccer has emerged as a vital aspect of sports analytics, largely due to the growing availability of extensive, annotated video datasets and significant advancements in deep learning technology. Accurately detecting and classifying fouls not only enhances the objectivity of match analysis but also provides essential support to referees during games. However, the inherent complexity of soccer matches—marked by rapid player movements, frequent occlusions, and a variety of foul scenarios—poses considerable challenges for traditional computer vision methods (Cioppa et al., 2020).

Recent developments in multi-view video analysis have demonstrated that synchronized camera feeds can significantly improve event recognition in sports (Gao et al., 2024). This technique allows for a richer understanding of player interactions and foul occurrences from multiple angles, ultimately enhancing the accuracy of detection systems. Despite these advancements, many current methods still face difficulties in effectively modeling long-range dependencies (Vaswani et al., 2023) and managing multiple tasks in real time (Carion et al., 2020). The fast-paced nature of soccer, characterized by quick transitions and intricate player formations, demands robust algorithms that can swiftly process large volumes of visual data.

Additionally, integrating context-aware loss functions has been suggested to refine action spotting in soccer videos, addressing the need for a more nuanced understanding of player actions and their implications (Cioppa et al., 2020). As the field continues to evolve, the creation of scalable datasets, such as Soccernet, is crucial for training models that can generalize effectively across various match scenarios (Giancola et al., 2018). Ultimately, the combination of advanced multi-view analysis, context-aware methodologies, and efficient deep learning architectures has the potential to transform foul recognition in soccer, making it more accurate and reliable for referees and analysts alike.

To address these challenges, we introduce `S-amba`, a novel multi-task framework tailored for

[a] https://orcid.org/0000-0003-0266-2465
[b] https://orcid.org/0000-0002-2786-2677
[c] https://orcid.org/0000-0003-0332-8231
[d] https://orcid.org/0009-0005-9635-7307

Velesaca, H. O., Gomez-Cantos, A., Reyes-Angulo, A. and Araujo, S.
S-amba: A Multi-View Foul Recognition in Soccer Through a Mamba-Based Approach.
DOI: 10.5220/0013682500003988

In *Proceedings of the 13th International Conference on Sport Sciences Research and Technology Support (icSPORTS 2025)*, pages 57-68
ISBN: 978-989-758-771-9; ISSN: 2184-3201

multi-view foul recognition in soccer, specifically designed for the `SoccerNet-MVFoul dataset`[1] (Held et al., 2023). Our approach harnesses the efficiency of the Mamba state space model, integrating a multi-task learning strategy to simultaneously predict foul occurrences, classify foul types, and localize key events across synchronized multi-view video inputs (Gu and Dao, 2024). The S-amba framework capitalizes on Mamba's ability to model long-range dependencies effectively, addressing the limitations of traditional methods in handling the dynamic and complex nature of soccer matches. Our main contributions are: (1) the S-amba architecture, which seamlessly processes synchronized multi-view video inputs, (2) a hybrid loss function that balances foul classification and event localization tasks, and (3) superior performance on the SoccerNet-MVFoul dataset (Held et al., 2023) compared to state-of-the-art models.

The manuscript is organized as follows. Section 2 introduces some related works for foul detection within the soccer context. Section 3 presents the proposed methodology carried out to implement the proposed architecture. Then, Section 4 shows the experimental results on a benchmark dataset. Finally, conclusions are presented in Section 5.

2 BACKGROUND

Automatic foul detection in sports events using computer vision techniques has gained significant relevance in recent years due to its potential to assist referees and improve decision-making accuracy ((Thomas et al., 2017)). Traditionally, approaches to sports action recognition have been based on deep learning methods, especially convolutional neural networks (CNNs) and recurrent networks (RNNs), which have proven effective in event classification and detection tasks in sports videos ((Carreira and Zisserman, 2017), (Feichtenhofer et al., 2019)).

However, most of these methods are limited to analysis from a single visual perspective, which restricts their ability to capture complete spatial and temporal information, especially in complex situations such as fault detection, where multiple views can provide crucial complementary information ((Iosifidis et al., 2013), (Putra et al., 2022)). Recently, multi-view approaches have emerged as a promising solution to overcome these limitations, integrating information from multiple views to improve the robustness and accuracy of action recognition (Shah et al., 2023). On the other hand, the work proposed by (Hu et al., 2008) present a method for recognizing facial expressions from multiple viewing angles. It addresses the variability in facial appearance, which makes emotion identification difficult. It uses image processing and machine learning techniques to combine information from different perspectives, thereby improving recognition accuracy.

Similar to the previous approach, the work presented by (Held et al., 2023) is the SoccerNet-MVFoul dataset, which contains multi-viewing angle videos of soccer fouls. It also presents an encoder-decoder architecture with a multitask classifier for foul and action recognition tasks. Furthermore, the proposed architecture has weaknesses, such as its dependence on high-resolution videos to obtain high accuracy values.

3 METHODOLOGY

This section details the different stages followed to carry out the proposed methodology in the context of Multi-view Foul Recognition (MVFR). The MVFR task involves classifying soccer videos from multiple camera views into foul categories (no offence, infraction severity 1, 3, or 5) and action types (e.g., standing tackling, tackling, holding, pushing, high leg, elbowing, dive, and challenge). This multi-task classification problem requires robust feature extraction, view aggregation, and task-specific predictions. A novel `MultiTask Model Mamba-based called S-amba`, which integrates a pre-trained MViT-V2-S backbone (Li et al., 2022) with an enhanced Mamba-based aggregation module (Gu and Dao, 2024), incorporating temporal and view attention mechanisms, as illustrated in Figure 1, to capture cross-view and temporal dependencies efficiently. Our approach addresses challenges such as class imbalance, noisy annotations, and multi-view integration through advanced preprocessing, curriculum learning, class-weighted loss functions, and gradual backbone unfreezing.

3.1 Model Architecture

The `S-amba model` processes multi-view video inputs of shape $\mathbf{x} \in \mathbb{R}^{B\times V\times C\times T\times H\times W}$, where B is the batch size, $V = 2$ is the number of views, $C = 3$ is the number of color channels, $T = 16$ is the number of frames, and $H \times W = 398 \times 224$ is the frame resolution. The model outputs logits for foul classification (4 classes) and action classification (8 classes). The architecture comprises a backbone, a Mamba-

[1] https://huggingface.co/datasets/SoccerNet/SN-MVFouls-2025

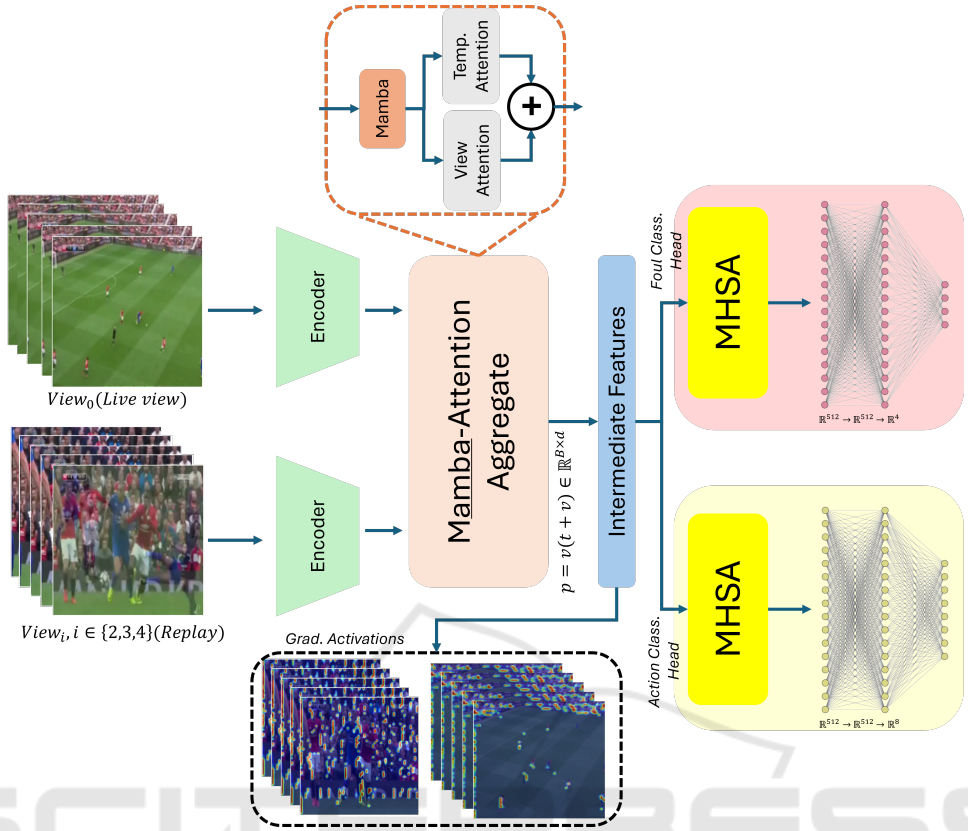

Figure 1: Overview of the proposed S-amba architecture for multi-view foul recognition. Each input action video consists of two synchronized views (live stream and a randomly selected replay), which are processed by a shared MViT-V2-S encoder. The extracted features are passed to the Mamba-Attention Aggregate module, which integrates long-range dependencies via a Mamba state space model and applies both temporal and view-level attention. The aggregated representation is then used to produce multi-task predictions for foul classification (4 classes) and action classification (8 classes) through attention-enhanced task-specific heads.

based view aggregation module with attention, and task-specific heads with attention mechanisms. Below, we describe each component mathematically.

3.1.1 Backbone: MViT-V2-S

The backbone is a pre-trained MViT-V2-S model (Li et al., 2022), initialized with Kinetics-400 weights (Kay et al., 2017). Input tensors are reshaped to $\mathbf{x}' \in \mathbb{R}^{(B\cdot V\cdot T)\times C\times H\times W}$ and resized to 224×224 via bilinear interpolation:

$$\mathbf{x}'' = \text{Interpolate}(\mathbf{x}', 224\times 224, \text{mode} = \text{bilinear}),$$

then reshaped back to $\mathbb{R}^{B\times V\times C\times T\times 224\times 224}$. The backbone extracts features as:

$$\mathbf{f} = \mathcal{F}_{\text{MViT}}(\mathbf{x}''; \theta_{\text{MViT}}) \in \mathbb{R}^{(B\cdot V)\times d}, \quad (1)$$

where $d = 512$ is the feature dimension, and θ_{MViT} are the backbone parameters. The original head is modified to a linear layer:

$$\mathbf{f} = \lambda_{\text{head}}(\mathbf{z}), \quad \mathbf{z} \in \mathbb{R}^{(B\cdot V)\times d_{\text{in}}}, \quad (2)$$

where λ_{head} projects to 512 dimensions. Initially, only the last two layers are unfrozen, with gradual unfreezing of additional layers every 2 epochs.

3.1.2 Mamba Aggregate

We introduce `Mamba Aggregate`, which enhances view aggregation using the Mamba state space model (Gu and Dao, 2024) combined with temporal and view attention mechanisms. Features $\mathbf{f}$ are reshaped to $\mathbf{f}' \in \mathbb{R}^{B\times V\times d}$:

$$\mathbf{f}' = \text{Unbatch}(\mathbf{f}, B, \text{dim} = 1, \text{unsqueeze} = \text{True}).$$

The Mamba module processes $\mathbf{f}'$ as a sequence of V views:

$$\mathbf{h}_t = \text{Mamba}(\mathbf{f}'_t, \mathbf{h}_{t-1}; \theta_{\text{Mamba}}) \in \mathbb{R}^{B\times d}, \quad t = 1,\ldots,V, \quad (3)$$

where $\mathbf{h}_t$ is the hidden state, and θ_{Mamba} are parameters with $d_{\text{state}} = 16$, $d_{\text{conv}} = 4$, and expansion factor 2. A lifting network processes the output:

$$\mathbf{m} = \mathcal{L}(\mathbf{h}_t; \theta_{\text{lift}}) \in \mathbb{R}^{B\times V\times d}, \quad (4)$$

where $\mathcal{L} = \nu \circ \sigma \circ \lambda_{d \to d}$, with σ as GELU activation, $\lambda_{d \to d}$ a linear transformation, and ν layer normalization. Temporal attention aggregates features across views:

$$\mathbf{t} = \mathcal{A}_{\text{temp}}(\mathbf{m}; \theta_{\text{temp}}) \in \mathbb{R}^{B \times d}, \quad (5)$$

where $\mathcal{A}_{\text{temp}}$ is a multi-head attention module with 4 heads, followed by mean pooling over the temporal dimension. View attention further processes features:

$$\mathbf{v} = \mathcal{A}_{\text{view}}(\mathbf{m}; \theta_{\text{view}}) \in \mathbb{R}^{B \times d}, \quad (6)$$

The final aggregated feature is:

$$\mathbf{p} = \nu(\mathbf{t} + \mathbf{v}) \in \mathbb{R}^{B \times d}. \quad (7)$$

3.1.3 Multi-Task Head

The aggregated features $\mathbf{p}$ are processed by a shared intermediate network:

$$\mathbf{g} = I(\mathbf{p}; \theta_{\text{inter}}) \in \mathbb{R}^{B \times d}, \quad (8)$$

where $I = \delta_{\text{path},0.1} \circ \delta_{0.5} \circ \sigma \circ \lambda_{d \to d} \circ \nu$, with σ as GELU, $\lambda_{d \to d}$ a linear transformation, ν layer normalization, $\delta_{0.5}$ dropout (probability 0.5), and $\delta_{\text{path},0.1}$ drop path (probability 0.1). Task-specific attention modules process view features $\mathbf{m} \in \mathbb{R}^{B \times V \times d}$:

$$\mathbf{f}_{\text{foul}} = \mathcal{A}_{\text{foul}}(\mathbf{m}; \theta_{\text{foul-attn}}) \in \mathbb{R}^{B \times d}, \quad (9)$$

$$\mathbf{f}_{\text{action}} = \mathcal{A}_{\text{action}}(\mathbf{m}; \theta_{\text{action-attn}}) \in \mathbb{R}^{B \times d}, \quad (10)$$

where $\mathcal{A}_{\text{foul}}$ and $\mathcal{A}_{\text{action}}$ are multi-head attention modules with 4 heads. Task-specific branches produce logits:

$$\mathbf{y}_{\text{foul}} = \mathcal{H}_{\text{foul}}(\mathbf{g} + \mathbf{f}_{\text{foul}}; \theta_{\text{foul}}) \in \mathbb{R}^{B \times 4}, \quad (11)$$

$$\mathbf{y}_{\text{action}} = \mathcal{H}_{\text{action}}(\mathbf{g} + \mathbf{f}_{\text{action}}; \theta_{\text{action}}) \in \mathbb{R}^{B \times 8}, \quad (12)$$

where $\mathcal{H}_{\text{foul}}$ and $\mathcal{H}_{\text{action}}$ are:

$$\mathcal{H} = \lambda_{d \to n} \circ \delta_{0.5} \circ \sigma \circ \lambda_{d \to d} \circ \nu,$$

with $n = 4$ for foul and $n = 8$ for action classification. The model outputs $(\mathbf{y}_{\text{foul}}, \mathbf{y}_{\text{action}})$.

3.2 Training Strategy

The model is trained on the SoccerNet-MVFoul dataset(Held et al., 2023), addressing class imbalance, noisy labels, and computational constraints using curriculum learning, class-weighted loss, data augmentation, and gradual unfreezing. Training is conducted on an A100 NVIDIA GPU using PyTorch (Paszke et al., 2019).

3.2.1 Dataset and Preprocessing

The dataset consists of video clips stored as `.pt` files, with input shape $[B, V, 3, 16, 398, 224]$. The `SoccerNet-MVFoul dataset` class normalizes pixel values to $[0, 1]$ and selects two views (live stream and replay, random or selected). Labels are mapped to foul classes (0: No Offence, 1: Severity 1, 2: Severity 3, 3: Severity 5) and action classes (0: Standing Tackling, 1: Tackling, 2: Holding, 3: Pushing, 4: Challenge, 5: Dive, 6: High Leg, 7: Elbowing). Invalid annotations (e.g., severities 2.0 or 4.0) are filtered, and action labels are normalized.

"Challenge": 4, "Dive": 5, "High Leg": 6, "Elbowing": 7

3.2.2 Loss Function

The `S-amba` framework employs a multi-task loss for foul prediction and action localization, tailored to the model variants in Table 1. For model v_1, we use class-weighted Cross-Entropy losses:

$$\mathcal{L}_{\text{v1}} = \mathcal{L}_{\text{foul}} + \mathcal{L}_{\text{action}}, \quad (13)$$

where:

$$\mathcal{L}_{\text{foul}} = -\sum_{i=1}^{B} w_{\text{foul},y_i} \log\left(p_{\text{foul},i}\right),$$

$$\mathcal{L}_{\text{action}} = -\sum_{i=1}^{B} w_{\text{action},z_i} \log\left(p_{\text{action},i}\right),$$

with $p_{\text{foul},i} = \text{Softmax}(\mathbf{y}_{\text{foul},i})[y_i]$, $p_{\text{action},i} = \text{Softmax}(\mathbf{y}_{\text{action},i})[z_i]$. For models v_2, v_3, v_4, and v_5, we apply Focal loss to handle class imbalance (Lin et al., 2017):

$$\mathcal{L}_{\text{v2-v5}} = \mathcal{L}_{\text{foul}}^{\text{focal}} + \mathcal{L}_{\text{action}}^{\text{focal}}, \quad (14)$$

where:

$$\mathcal{L}_{\text{foul}}^{\text{focal}} = -\sum_{i=1}^{B} w_{\text{foul},y_i} (1 - p_{\text{foul},i})^{\gamma} \log\left(p_{\text{foul},i}\right),$$

$$\mathcal{L}_{\text{action}}^{\text{focal}} = -\sum_{i=1}^{B} w_{\text{action},z_i} (1 - p_{\text{action},i})^{\gamma} \log\left(p_{\text{action},i}\right),$$

with $\gamma = 2$. Class weights are computed as:

$$w_k = \frac{1}{\sqrt{n_k + \varepsilon}}, \quad \varepsilon = 10^{-6}, \quad (15)$$

where n_k denotes class counts, normalized to sum to 1. Label smoothing (0.05) is applied across all variants to reduce overfitting (Szegedy et al., 2016).

3.2.3 Optimization

We use AdamW (Loshchilov and Hutter, 2017) with an initial learning rate of 5×10^{-5} and weight decay 0.01. A OneCycleLR scheduler (Smith and Topin, 2019) adjusts the learning rate to a maximum of 1×10^{-4}. The backbone starts with the last two layers unfrozen, with one additional layer unfrozen every 2 epochs. Gradient clipping (max norm 1.0) prevents exploding gradients.

3.2.4 Curriculum Learning

To address class imbalance, we over-sample rare classes (e.g., Dive, High Leg, Elbowing). For each sample i, the sampling factor is:

$$\text{factor}_i = \min(\max(\alpha_{\text{action},i}, \alpha_{\text{foul},i}), 50), \tag{16}$$

where:

$$\alpha_{\text{action},i} = \begin{cases} 3 & \text{if class is Dive,} \\ 2 & \text{if class is High Leg or Elbowing,} \\ 1 & \text{otherwise,} \end{cases}$$

$$\alpha_{\text{foul},i} = \begin{cases} 5 & \text{if class is No Offence or Severity 5,} \\ 2 & \text{if class is Severity 3,} \\ 1 & \text{otherwise.} \end{cases}$$

This increases the effective training set size, as reported in the training script.

3.2.5 Data Augmentation

Training clips undergo random augmentations using Kornia (Riba et al., 2020):

- Random horizontal flip ($p = 0.5$).
- Random affine transform (rotation $\pm 15^\circ$, translation $\pm 10\%$, scale $[0.8, 1.2, 1.5, 2.0]$).
- Color jitter (brightness, contrast, saturation ± 0.3, hue ± 0.1, $p = 0.5$).

3.2.6 Evaluation and Early Stopping

The model is evaluated using accuracy and balanced accuracy (BA):

$$BA = \frac{1}{K} \sum_{k=1}^{K} \frac{\text{TP}_k}{\text{TP}_k + \text{FN}_k}, \tag{17}$$

where $K = 4$ for foul and $K = 8$ for action classification. Early stopping is triggered if the combined BA does not improve for 50 epochs.

To provide a comprehensive assessment of the model's performance, we report several key metrics: top-1 accuracy ($Acc.@1$), top-2 accuracy ($Acc.@2$), F1-score ($F1$), recall (RE), and precision (PR). Top-1 accuracy indicates the proportion of instances where the model's highest-ranked prediction matches the ground truth. Top-2 accuracy considers a prediction correct if the true label is among the model's two highest-ranked outputs, which is particularly informative in cases of label ambiguity or when multiple plausible labels exist.

Because the dataset has a significant class imbalance, just looking at accuracy alone can be misleading. A model might get a high accuracy score simply by favoring the most common classes. That's why we're also including these other metrics, which give us a more detailed understanding of performance:

$$\text{Precision } (PR) = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{18}$$

$$\text{Recall } (RE) = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{19}$$

$$\text{F1-score } (F1) = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{20}$$

where TP, FP, and FN stand for true positives, false positives, and false negatives, respectively. The F1-score, which is the harmonic mean of precision and recall, is particularly helpful for imbalanced datasets. It balances the trade-off between making false positive errors and missing actual positive cases.

By using all these metrics, we can be sure we're getting a fair and thorough evaluation of our model. This approach helps us understand not only how well it performs overall, but also how well it identifies less common classes and how reliable its predictions are. You'll find the specific numbers for each metric in the next section.

3.3 Implementation Details

The model is trained for up to 200 epochs with a batch size of 16 on an A100 NVIDIA GPU. The training and validation datasets follow the SoccerNet structure (Giancola et al., 2018). Predictions and ground truth are logged in JSON format. Compared to traditional Transformer-based approaches (Vaswani et al., 2017), the S-amba implementation reduces memory overhead, enabling efficient processing of synchronized multi-view inputs. The input video frames are processed at resolutions of 224×398 and 112×199 to balance computational cost and model performance. Table 1 details the experimental configurations for all conducted experiments.

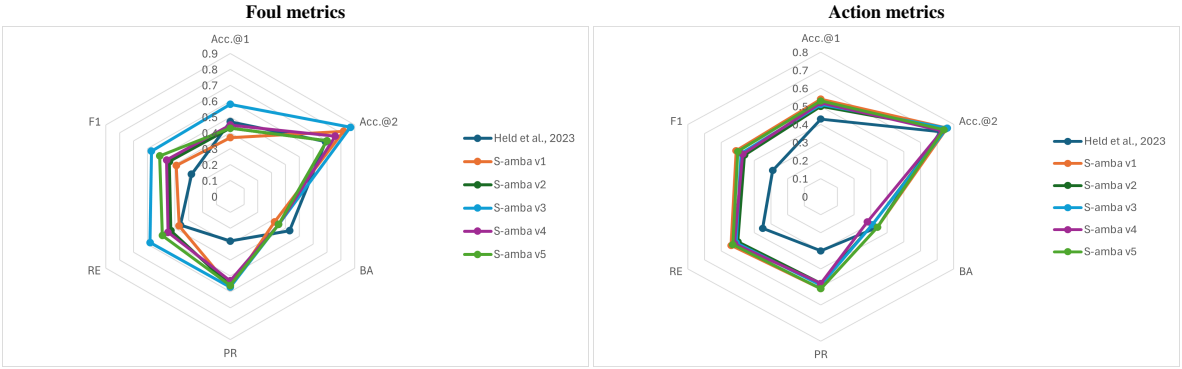

Figure 2: Radial graphs for foul and action classification.

Table 1: Experiment setup. Live Stream (L), Replay (R).

Model	Views	Strategy
(Ours) v_1	$L+R_{Random}$	Unfreeze all backbone
(Ours) v_2	$L+R_{Random}$	Unfreeze 2 last layers
(Ours) v_3	$L+R_1$	Unfreeze 4 last layers
(Ours) v_4	R_1+R_2	Unfreeze 4 last layers
(Ours) v_5	$L+R_1$	Gradual unfreeze

4 EXPERIMENTAL RESULTS

The `S-amba models` are evaluated on the SoccerNet-MVFoul test dataset (Held et al., 2023), comparing them with CNN- and Transformer-based approaches. The evaluation results are summarized in Tables 2 and 3, and a visual representation of these metrics with radial graphs is shown in Figure 2. On the other hand, Figures 3 show the confusion matrices of the different model evaluations.

4.1 Dataset

An exploratory analysis of the dataset is conducted as a first step, revealing a clear imbalance in the distribution of classes, both in terms of foul presence (offense/no offense) and the severity levels assigned to each event. As shown in Figure 4, Most examples fall into the ”no offense” category with low severity (-1), while foul events (”offense”) are mainly distributed across intermediate severities (1 and 3). Cases with high severity (4 and 5) are very rare. The ”between” class, which represents ambiguous cases, is also mostly concentrated in the low and medium severity levels.

This distribution highlights the unbalanced nature of the problem, where the majority classes can dominate the model's learning process, making it harder to correctly identify less frequent but important events, such as serious fouls. That's why it's essential to use evaluation metrics that reflect performance across all classes, not just the most common ones.

On the other hand, the relationship between event type (challenge/offense) and severity is crucial for automatic foul recognition. Events labeled as ”offense” tend to be associated with higher severity levels, while ”no offense” events are grouped at the lower end of the scale. This correlation suggests that severity could be a useful indicator for classifying and prioritizing events within the context of the challenge.

Additionally, Figure 5 shows the joint distribution between action classes and severity levels. We can see that certain actions, such as ”dive” and ”unknown,” are almost exclusively found at the lowest severity levels (-1), while other actions like ”elbowing,” ”high leg,” ”pushing,” and ”tackling” are spread across a wider range of severities, including intermediate and high values.

This matrix makes it clear that severity is not distributed evenly among the different action classes. Actions like ”elbowing” and ”high leg” tend to be associated with higher severity, suggesting that severity could be a useful and discriminative attribute for classifying dangerous or sanctionable actions. In contrast, actions like ”dive” and ”unknown” are rarely linked to high severity, which may reflect both the nature of these actions and possible ambiguities in the annotations. The relationship between action class and severity underscores the importance of considering both variables when designing foul recognition models, as this allows for prioritizing the detection of events that have a greater impact on the game.

4.2 Foul Classification

After performing the dataset analysis, the next step is to present the analysis results for foul and action classification. For foul classification, the `S-amba model v3` generally performs better in the $Accmetrics.@1 =$

Table 2: Test set performance for the multi-view video foul and action classification. Balanced Accuracy (*BA*), F1-score ($F1$), Recall (*RE*), Precision (*PR*).

Type	Author	Feature Extractor	Pooling	Size	*Acc.*@1	*Acc.*@2	*BA*	*PR*	*RE*	$F1$
Foul	(Held et al., 2023)	ResNet (He et al., 2016)	Mean	224×398	0.32	0.60	0.28	-	-	-
Foul	(Held et al., 2023)	ResNet (He et al., 2016)	Max	224×398	0.32	0.60	0.28	-	-	-
Foul	(Held et al., 2023)	R(2+1)D (Tran et al., 2018)	Mean	224×398	0.32	0.56	0.33	-	-	-
Foul	(Held et al., 2023)	R(2+1)D (Tran et al., 2018)	Max	224×398	0.32	0.56	0.33	-	-	-
Foul	(Held et al., 2023)	MViT-V2-S (Li et al., 2022)	Mean	224×398	0.40	0.65	**0.45**	-	-	-
Foul	(Held et al., 2023)	MViT-V2-S (Li et al., 2022)	Max	224×398	0.47	0.69	0.43	0.28	0.36	0.28
Foul	`S-amba v1` (Ours)	MViT-V2-S (Li et al., 2022)	-	224×398	0.37	0.82	0.32	**0.57**	0.37	0.39
Foul	`S-amba v2` (Ours)	MViT-V2-S (Li et al., 2022)	-	112×199	0.43	0.70	0.35	0.56	0.43	0.44
Foul	`S-amba v3` (Ours)	MViT-V2-S (Li et al., 2022)	-	112×199	**0.58**	**0.87**	0.35	**0.57**	**0.58**	**0.57**
Foul	`S-amba v4` (Ours)	MViT-V2-S (Li et al., 2022)	-	112×199	0.45	0.76	0.35	0.53	0.45	0.46
Foul	`S-amba v5` (Ours)	MViT-V2-S (Li et al., 2022)	-	112×199	0.43	0.70	0.35	0.56	0.49	0.51
Action	(Held et al., 2023)	ResNet (He et al., 2016)	Mean	224×398	0.34	-	0.25	-	-	-
Action	(Held et al., 2023)	ResNet (He et al., 2016)	Max	224×398	0.32	-	0.24	-	-	-
Action	(Held et al., 2023)	R(2+1)D (Tran et al., 2018)	Mean	224×398	0.34	-	0.30	-	-	-
Action	(Held et al., 2023)	R(2+1)D (Tran et al., 2018)	Max	224×398	0.39	-	0.31	-	-	-
Action	(Held et al., 2023)	MViT-V2-S (Li et al., 2022)	Mean	224×398	0.38	-	0.31	-	-	-
Action	(Held et al., 2023)	MViT-V2-S (Li et al., 2022)	Max	224×398	0.43	0.72	**0.34**	0.30	0.35	0.29
Action	`S-amba v1` (Ours)	MViT-V2-S (Li et al., 2022)	-	224×398	**0.54**	**0.76**	**0.34**	**0.51**	**0.54**	**0.51**
Action	`S-amba v2` (Ours)	MViT-V2-S (Li et al., 2022)	-	112×199	0.50	0.75	0.31	0.48	0.50	0.46
Action	`S-amba v3` (Ours)	MViT-V2-S (Li et al., 2022)	-	112×199	0.51	**0.76**	0.31	0.49	0.51	0.48
Action	`S-amba v4` (Ours)	MViT-V2-S (Li et al., 2022)	-	112×199	0.52	0.73	0.28	0.48	0.52	0.47
Action	`S-amba v5` (Ours)	MViT-V2-S (Li et al., 2022)	-	112×199	0.53	0.74	**0.34**	**0.51**	0.53	0.50

Table 3: Test set performance for the multi-view video foul and action classification.

Foul	*Acc.*@1				
	v1	**v2**	**v3**	**v4**	**v5**
No Offence	38.10	42.86	28.57	47.62	**52.38**
Offence Severity 1	29.41	36.31	**71.34**	44.59	59.24
Offence Severity 3	**61.76**	**61.76**	39.71	47.06	27.94
Offence Severity 5	0.00	0.00	0.00	0.00	0.00
Action	*Acc.*@1				
	v1	**v2**	**v3**	**v4**	**v5**
Standing Tackling	77.42	80.37	76.64	**84.11**	80.37
Tackling	**68.75**	44.18	51.16	39.53	44.19
Holding	3.45	7.14	10.71	7.14	**17.86**
Pushing	**40.00**	0.00	0.00	0.00	0.00
Challenge	29.79	21.74	26.09	**34.78**	30.43
Dive	0.00	0.00	**20.00**	0.00	0.00
High Leg	**33.33**	**33.33**	16.67	**33.33**	**33.33**
Elbowing	16.67	**63.64**	54.55	27.27	**63.64**

0.58, *Acc.*@2 = 0.87, *PR* = 0.57, *RE* = 0.58, $F1$ = 0.57, and only being surpassed in *BA* = 0.35 by (Held et al., 2023) with MViT-V2-S in Table 2 (*5th row*). On the other hand, the radial graph shown in Figure 2 (*1st col*) graphically shows the quantitative result of the metrics analyzed and mainly highlights `S-amba model v3`. The confusion matrix present in Figure 3 (*1st col*) shows that the model tends to incorrectly classify Severity 5 offenses as Severity 1 or Severity 3 in major cases, suggesting that the model captures the presence of serious offenses but struggles to distinguish between extreme severity levels, likely due to the scarcity of Severity 5 examples in the training data.

The class analysis reveals different patterns shown above:

- **No Offense:** Accuracy of 52.38%, showing moderate performance in identifying fair plays.
- **Offense Severity 1:** With accuracy of 71.34%, being the category best recognized by the model and reflecting the excellent performance of detecting minor offenses.
- **Offense Severity 3:** Accuracy of 61.76%, reflecting the good performance of detecting moderate offenses.
- **Offense Severity 5:** Accuracy of 0.00%, indicating a significant limitation in identifying the most severe but extremely rare offenses in the dataset.

4.3 Action Classification

The next task to analyze the results is action classification, where the `S-amba model v1` achieves the metrics value as *Acc.*@1 = 0.54, *Acc.*@2 = 0.76, *BA* = 0.34, *PR* = 0.51, *RE* = 0.54, and $F1$ = 0.51, significantly outperforming the different models. On the other hand, the radial graph (Figure 2) (*2nd column*) graphically shows that the `S-amba model v1, v2, v3, and v5` present very close values. The confusion matrix (Figure 3) (*2nd col*) reveals that the model frequently confuses Elbowing with Challenge; Standing Tackling with Challenge, Holding and Tackling, which is understandable given the visual similarity between these actions and the class imbalance.

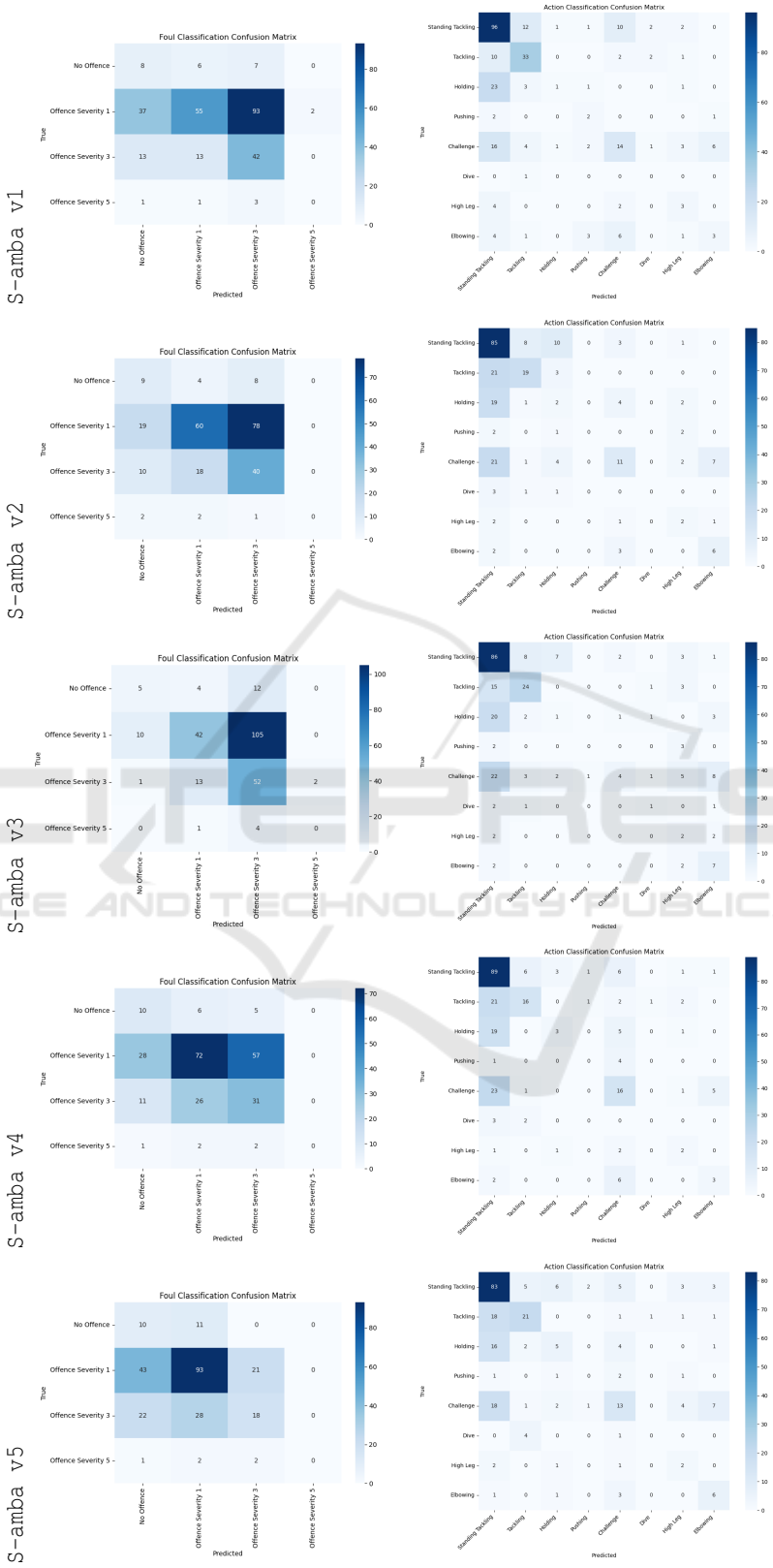

Figure 3: *1st col.* Confusion matrix for foul classification. *2nd col.* Confusion matrix for action classification.

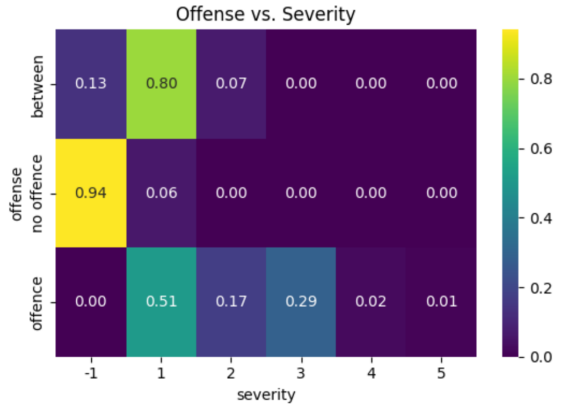

Figure 4: Joint distribution of offense and severity in the dataset.

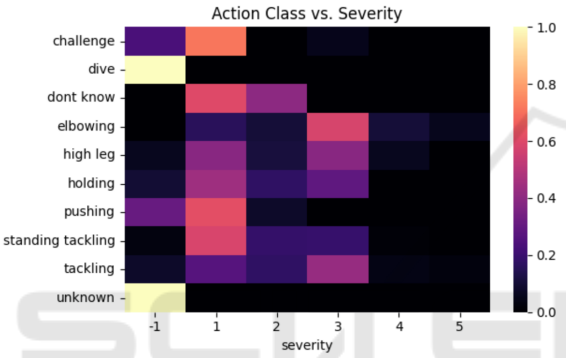

Figure 5: Distribution of action classes based on severity.

Performance by class shows important observations:

- **Standing Tackling:** Accuracy of 84.11%, being the best-recognized action.
- **Tackling:** The model achieved an accuracy of 68.75%, which reflects its effectiveness in identifying dynamic contact actions within the dataset.
- **Elbowing:** An accuracy of 63.64% was obtained, indicating the model's capability to correctly identify this category of action.
- **Pushing:** A accuracy of 40.00%, which may be attributed to the limited number of instances of this action in the dataset.
- **Challenge, High Leg, and Holding:** Accuracies of 34.78%, 33.33%, and 17.86%, respectively, reflecting the difficulty of recognizing less frequent actions.
- **Dive:** Accuracy of 20.00%, demonstrating the extreme difficulty of detecting malingering, which is both rare and visually subtle.

Figure 6 shows the feature visualization of feature maps obtained from the proposed architecture, evaluated on sample frames from the test set. These visualizations reveal that the model pays special attention to regions of interaction between players, highlighting areas of potential contact. Additionally, the Mamba-based aggregation effectively captures complementary information from multiple views, integrating spatial features from different angles. Notably, the activations are strongest in frames containing the exact moment of the foul, demonstrating the model's ability to locate relevant events temporally.

Figure 7 shows Grad-CAM visualizations obtained from the proposed architecture evaluated on test set samples from two different views. It is observed that, in both samples, the first layers capture low-level features with sparse activations. In contrast, in deeper layers, the activations become more focused, concentrating on the regions of interest. This progressive evolution of the activations evidences an effective hierarchical representation capability, and the consistency between views suggests a robust generalization of the model to perspective changes.

4.4 Reproducibility

The codebase will be available at GitHub and includes all scripts for preprocessing (`preproc.py`), model definition (`model_2.py`), and training (`train_mamba.py`). Instructions for setup and execution are provided in the repository's README. We also save predictions and ground truth JSON files for verification.

5 CONCLUSIONS

The `S-amba` multi-task architecture effectively tackles Multi-View Foul Recognition using the SoccerNet-MVFoul dataset (Held et al., 2023), by leveraging sequential modeling for multi-view aggregation alongside robust training strategies to handle imbalanced data. The results demonstrate that the proposed `S-amba` architecture outperforms competing methods across key metrics such as $Acc.@1$, $Acc.@2$, PE, RE, and $F1$, with the only exception being BA in foul classification, where it is slightly surpassed. In contrast, for action classification, `S-amba` achieves superior performance across all evaluated metrics. Notably, the architecture employs MViT-V2-S as its backbone and utilizes Mamba with a video input size of 112×199—half the size used by other architectures—yet still delivers the best results for multi-view foul and action classification in video. This surpasses previous approaches proposed by (Held et al., 2023). Future work will focus on exploring larger backbone

Figure 6: Feature map visualizations from the proposed architecture evaluated on test set sample frames.

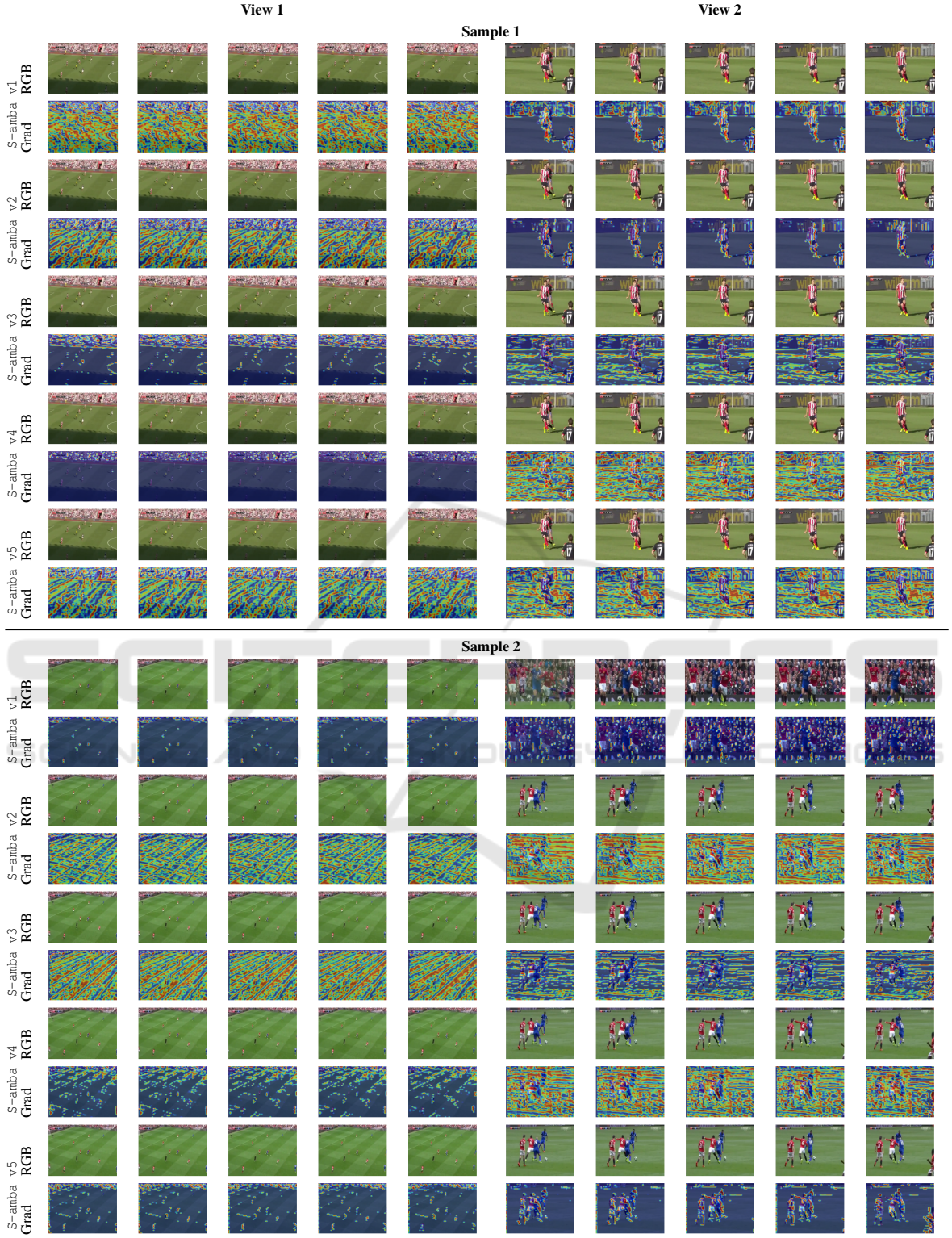

Figure 7: Grad-CAM visualizations from the proposed architecture evaluated on test set sample frames.

models and advanced data augmentation techniques to further enhance performance.

ACKNOWLEDGEMENTS

This research has been supported by the ESPOL project “Reconocimiento de patrones en imágenes usando técnicas basadas en aprendizaje”.

REFERENCES

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. (2020). End-to-end object detection with transformers.

Carreira, J. and Zisserman, A. (2017). Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308.

Cioppa, A., Deliège, A., Giancola, S., Ghanem, B., Droogenbroeck, M. V., Gade, R., and Moeslund, T. B. (2020). A context-aware loss function for action spotting in soccer videos.

Feichtenhofer, C., Fan, H., Malik, J., and He, K. (2019). Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211.

Gao, Y., Lu, J., Li, S., Li, Y., and Du, S. (2024). Hypergraph-based multi-view action recognition using event cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(10):6610–6622.

Giancola, S., Amine, M., Dghaily, T., and Ghanem, B. (2018). Soccernet: A scalable dataset for action spotting in soccer videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 1711–1721.

Gu, A. and Dao, T. (2024). Mamba: Linear-time sequence modeling with selective state spaces.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Held, J., Cioppa, A., Giancola, S., Hamdi, A., Ghanem, B., and Van Droogenbroeck, M. (2023). Vars: Video assistant referee system for automated soccer decision making from multiple views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5086–5097.

Hu, Y., Zeng, Z., Yin, L., Wei, X., Zhou, X., and Huang, T. S. (2008). Multi-view facial expression recognition. In *2008 8th IEEE International Conference on Automatic Face & Gesture Recognition*, pages 1–6. IEEE.

Iosifidis, A., Tefas, A., and Pitas, I. (2013). Multi-view human action recognition: A survey. In *2013 Ninth international conference on intelligent information hiding and multimedia signal processing*, pages 522–525. IEEE.

Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al. (2017). The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*.

Li, Y., Wu, C.-Y., Fan, H., Mangalam, K., Xiong, B., Malik, J., and Feichtenhofer, C. (2022). Mvitv2: Improved multiscale vision transformers for classification and detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4804–4814.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. (2017). Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.

Loshchilov, I. and Hutter, F. (2017). Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32.

Putra, P. U., Shima, K., and Shimatani, K. (2022). A deep neural network model for multi-view human activity recognition. *PloS one*, 17(1):e0262181.

Riba, E., Mishkin, D., Ponsa, D., Rublee, E., and Bradski, G. (2020). Kornia: an open source differentiable computer vision library for pytorch. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3674–3683.

Shah, K., Shah, A., Lau, C. P., de Melo, C. M., and Chellappa, R. (2023). Multi-view action recognition using contrastive learning. In *Proceedings of the ieee/cvf winter conference on applications of computer vision*, pages 3381–3391.

Smith, L. N. and Topin, N. (2019). Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*, volume 11006, pages 369–386. SPIE.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.

Thomas, G., Gade, R., Moeslund, T. B., Carr, P., and Hilton, A. (2017). Computer vision for sports: Current applications and research topics. *Computer Vision and Image Understanding*, 159:3–18.

Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., and Paluri, M. (2018). A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2023). Attention is all you need.