Technical Report for SoccerNet Multi-View Foul Recognition Challenge 2024

Milosz Lopatto
Faculty of Electronics and Information Technology
Warsaw University of Technology
Warsaw, Poland
milosz.lopatto@gmail.com

Adam Gorski
Faculty of Electronics and Information Technology
Warsaw University of Technology
Warsaw, Poland
adamsebastiangorski@gmail.com

***Abstract*—This document is technical report for SoccerNet Multi-View Foul Recognition Challenge 2024. It shortly describes the key changes we have made to the original VARS model in order to beat the baseline. We have participated as a team named *PW ZZSN*.**

I. Implementation

We have based our solution on the existing *VARS* model with the *MViTv2* encoder. First, we refactored and wrapped the code in **Pytorch Lightning**, which allowed us to easily track experiments using a custom logger for **Weights and Biases**.

II. Model configuration

```
pre_model:"mvit_v2_s"
pooling_type:"attention"
num_views:5
fps:12
start_frame:58
end_frame:92
batch_size:2
max_num_worker:4
LR:0.00005
gamma:0.3
step_size:3
weight_decay:0.001
data_aug:true
weighted_loss:true
```

A. Frames per second

We have experimented with multiple parameters, but fps, start frame and the end frame are the ones that have been changed compared to the baseline model configuration. As suggested in the original VARS model repository, we have kept the middle of the extracted clip in the 75th frame. However, we have experimented with different numbers of frames per second, and eventually we have left it at **12 FPS** instead of 17 FPS proposed in the baseline model.

III. Data augmentation

We have also been experimenting with the data augmentation. Besides making existing transformations more aggressive, we have also added **RandomResizedCrop** and **GaussianBlur**.

For the best model, we used the following data augmentation.

```
transformAug = transforms.Compose([
    transforms.RandomResizedCrop(
        size=(224, 224),
        scale=(0.8, 1.0)
    ),
    transforms.RandomAffine(
        degrees=(0, 0),
        translate=(0.2, 0.2),
        scale=(0.8, 1.2)
    ),
    transforms.RandomPerspective(
        distortion_scale=0.5,
        p=0.5
    ),
    transforms.RandomRotation(degrees=10),
    transforms.ColorJitter(
        brightness=0.6,
        saturation=0.6,
        contrast=0.6
    ),
    transforms.RandomHorizontalFlip(),
    transforms.GaussianBlur(
        kernel_size=(5, 9),
        sigma=(0.1, 5)
    ),
])
```

IV. Model training

This specific training was defined for 20 epochs. It took 4h 5min on NVIDIA A100 GPU. The final predictions were taken from the 12th epoch version of the model.

V. Future ideas: encoder selection

We have experimented with different encoders, but ultimately the *MViTv2* used in the baseline model turned out to be the best. If we had more time we would try the ***Hiera*** architecture [1] and see how it compares to the *MViTv2*.

VI. Code

At the moment the repository is private, but eventually we will share it after some cleaning under the following link: *https://github.com/milosz-l/multi-view-foul-recognition*

References

[1] C. Ryali, Y.-T. Hu, D. Bolya, C. Wei, H. Fan, P.-Y. Huang, V. Aggarwal, A. Chowdhury, O. Poursaeed, J. Hoffman, J. Malik, Y. Li, and C. Feichtenhofer, "Hiera: A Hierarchical Vision Transformer without the Bells-and-Whistles," arXiv:2306.00989, 2023. [Online]. Available: https://arxiv.org/abs/2306.00989