HIGH SCHOOL EDITION
Journal of Student Research
Volume 12 Issue 1 (2023)

Automatically Labeling Offensive Formations in American Football Film Using Deep Learning

Kyle Zhou[1] and Jason Galbraith[1#]

[1]Sunset High School, Portland, OR, USA
[#]Advisor

ABSTRACT

Web services for storing, annotating, and sharing sports videos by high-school athletic teams have become more prevalent in recent years. However, most services lack the ability for coaches to automatically tag film, leading to many hours of manual annotation. For American football videos, coaches need to label formations, plays, and field positions in order to extract insights and create strategic game plans. This paper presents an end-to-end machine learning pipeline for automatically labeling American football offensive formations in videos. The pipeline includes pre-processing of videos, image classification, and a novel inference approach. The study compares a custom CNN model with pre-trained image classifier models using transfer learning. Specifically, CNN-based architectures (MobileNet, Inception, EfficientNet, etc.) and a transformer-based Vision Transformer (ViT) are compared. All models are trained on ~1400 images with the three most popular formation labels extracted from video clips of high-school football team games. The results show that several models, including the custom CNN model, achieved greater than 90% classification accuracy on the test dataset. The inference is performed by sampling multiple frames from a video clip, passing them through a trained image classifier, and taking a majority vote on the classification results to determine the final outcome. Our study found that using a sampling rate of 0.5 seconds, starting at 1 second, and taking five frames yields the highest inference accuracy of 95.4% using a trained customized CNN model. This system can assist all levels of football coaches in the analysis of game footage and formation identification.

Introduction

High school football coaches often analyze extensive collections of game film to discover patterns in opponent play and create a corresponding strategic decision. However, before these insights can be found, coaches must annotate hours of play clips to provide context for these patterns. During annotation, it is natural for humans to commit errors because of the repetitiveness and sheer size of video data.

In football, there are two main types of plays: offense and defense. In this paper, we focus on one crucial feature of every offensive/defensive play: offensive formation. A formation is the configuration that the offensive players line up in before a play starts. The type of formation usually determines the type of plays the offense will run and also how the defensive team will align against the offense. Detection of teams' offensive formation tendencies enables further analysis in play detection and strategic recommendations.

Previous papers have done work on classifying offensive football formations; however, these studies do not achieve a high enough accuracy to replace manual annotation and mostly use traditional machine-learning techniques instead of deep neural network-based approaches. One paper in this area is (Ajmeri & Shah, 2018), in which researchers used five models (Classification and Regression Trees, Naive Bayes, SVM, K Neighbors, and Logistic Regressions) to classify the quarterback position and formation. The researchers' best-performing model used classification and regression trees to classify 29 total formations with a 72.3% accuracy and quarterback position with an 86.5% accuracy. Another paper (Atmosukarto et al., 2013) approaches the

problem using an SVM classifier to classify three top-level formations and eight second-level labels. They classify their top-level labels with a 67.1% accuracy and achieve an overall classification accuracy of 31.1% for all eight classes using a multi-class classifier. This approach requires manual feature extraction and heavy pre-processing, which is tedious and not scalable. Ultimately, both approaches do not utilize SOTA techniques, such as CNNs (Li et al., 2022) or Transformers (Vaswani et al., 2017), which have demonstrated significantly higher accuracy on image/video classification tasks.

CNNs (Convolutional Neural Networks) are deep learning neural networks that are particularly effective at image classification tasks. They have become the dominant technology in this field due to their ability to learn hierarchical features from input data, making them highly effective at identifying patterns and features within images. Additionally, CNNs can learn from large amounts of data, which is essential for achieving high levels of accuracy in image classification tasks. A transformer is another breakthrough neural network architecture introduced in 2017 and has since dominated the field of natural language processing (NLP). It is based on the concept of self-attention, which allows the model to focus on different parts of the input sequence and weigh their importance. This allows transformers to achieve parallelization within the model, making it highly efficient at processing long data sequences. It has also been applied to other domains beyond NLP, such as computer vision, with promising results. These advancements can certainly improve sports analytics, including the ability to automatically label football formations more accurately.

In this study, we designed a three-stage machine learning pipeline including video clip pre-processing, image classification model training, and a novel inference approach. In addition to building a customized CNN-based image classifier and training it from scratch, we also explored a transfer learning technique that uses a pre-trained image classification model as a feature extractor and trains the appended final layers on top. TensorFlow hub includes a rich set of pre-trained classifiers as well as corresponding feature extractors. We examined the most popular CNN-based architectures such as MobileNet (Howard et al., 2017), Inception (Szegedy et al., 2015), ResNet (He et al. 2016), and EfficientNet (Tan et al., 2019). We also compared the transformer-based ViT (Dosovitskiy et al., 2020) model with others in terms of the total number of model parameters, training time using GPU, and test accuracy. While the customized CNN model achieves 94% accuracy on the test dataset, EfficientNet gets the highest accuracy of 96% followed by MobileNet at 93%. The ViT model's 89% accuracy may be due to insufficient training data. The inference is performed by taking multiple frames, sampled at a fixed rate of 0.5 seconds, from a video clip, feeding them into a trained classifier, and using a majority vote to determine the outcome. Our study shows that properly selecting the number of frames and the time window for sampling can improve performance compared to using only one frame from the video clip.

Methods

Datasets

As with any supervised machine learning task, acquiring high-quality labeled data is key to success. There were few ways previous studies were able to compile a dataset. In one study, the authors collected formation images from the Madden NFL 2020 PC game (Newman, 2022) and another study (Atmosukarto et al., 2013) complied their own dataset of 815 distinct videos combined with 51 formation images from the Oregon State University Digital Scout Project (Hess, Fern, & Mortensen, 2007).

Football formations are defined at the start of each play and the plays are captured in separate video clips. In our study, we started with three popular formations with the greatest number of video clips that are downloaded from the local football team video repository, including the school district football games for the past few years. Table 1 lists the number of video clips for each of the three formations. Those video clips have been analyzed and properly labeled through the Hudl Assist service. Figure 1 shows some images sampled from video clips for those three formations (Trio, Empty, and Doubles).

Table 1. Three Football Formations and Corresponding Number of Video Clips.

Formations	Trio	Doubles	Empty
Number of Videos	240	203	56

Machine Learning Pipeline

Figure 2 shows the proposed pipelines based on deep-learning neural network architectures. The pipeline includes video pre-processing, training, and inference.

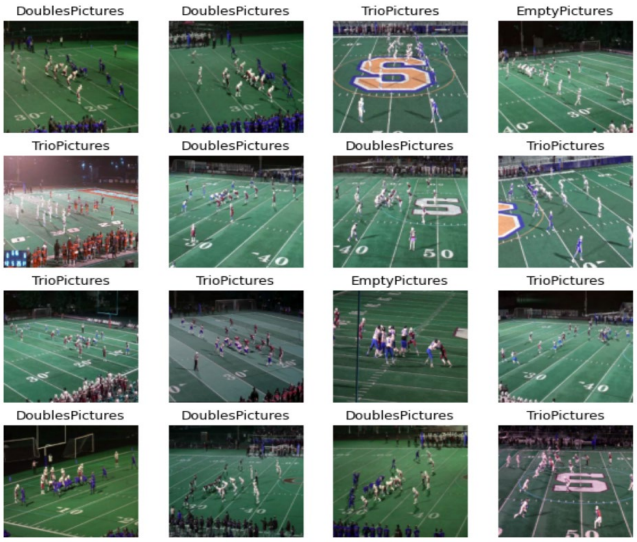

Figure 1. Football Formation Images Sampled from Video Clips

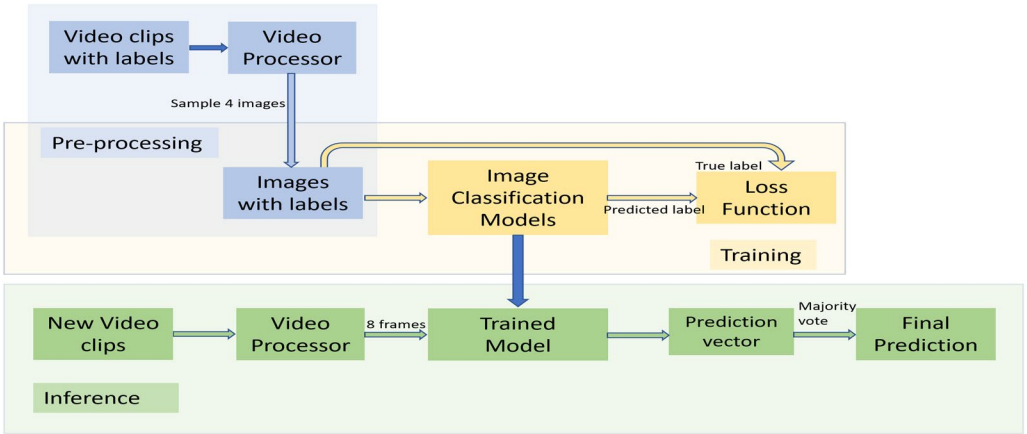

Figure 2. Machine Learning Pipeline for Football Formations Classification

Video Pre-Processing

For each video clip, four frames are sampled from 1s to 3s at a half-second sampling rate to capture the start of each play. Every frame is resized to 224x224 in order to match the input format of the downstream image classification model. A Python function using OpenCV libraries is written to implement the video to frame sampling and resizing.

The generated frames and the corresponding formation label are put under the corresponding directory. The TensorFlow utility function (tf.keras.utils.image_dataset_from_directory) is used to create the dataset that can be efficiently used for training the models. The dataset is partitioned into batches where each batch includes 32 randomly sampled images. It is standard to split the data into train, validation, and test where validation data can be used to fine-tune hyperparameters such as learning rate, model architectures, etc while the test dataset is reserved to report the final accuracy at the end. We split the data into three datasets using a 70/10/20 ratio.

Training Process

The training stage starts with building the image classification models using TensorFlow. Next, we define the loss function using cross entropy for classification. We use the popular Adam optimizer to train the model for 40 epochs. Plotting the loss and accuracy curve allows us to spot whether the selected model overfits. Several techniques such as dropout, data augmentation, and regularization can be deployed to counter the overfit if needed. Those architecture selections and modifications are done iteratively using the validation dataset. In the end, the trained model is used on the test dataset to report classification accuracy.

Inference

At deployment, the new video clip goes through the video processing to sample N frames every half second starting at T seconds. The final formation prediction is the majority vote of N model predictions for each frame where N and T are hyperparameters. This novel idea makes prediction robust to a timing shift of video clip cuts. For example, Figure 3 shows the model prediction and how the majority vote correctly labels the video as Doubles formation even though the first and fifth frames were wrongly predicted as “Trio”. In this case, N is set to eight and T is set to 0 seconds.

```
8 frames prediction index: [2 0 0 0 2 0 0 0]
8 frames prediction: ['Reo', 'Doubles', 'Doubles', 'Doubles', 'Reo', 'Doubles', 'Doubles', 'Doubles']
prediction of the formation is : Doubles
```

Figure 3. Example Inference for Video Clip.

Image Classification Models

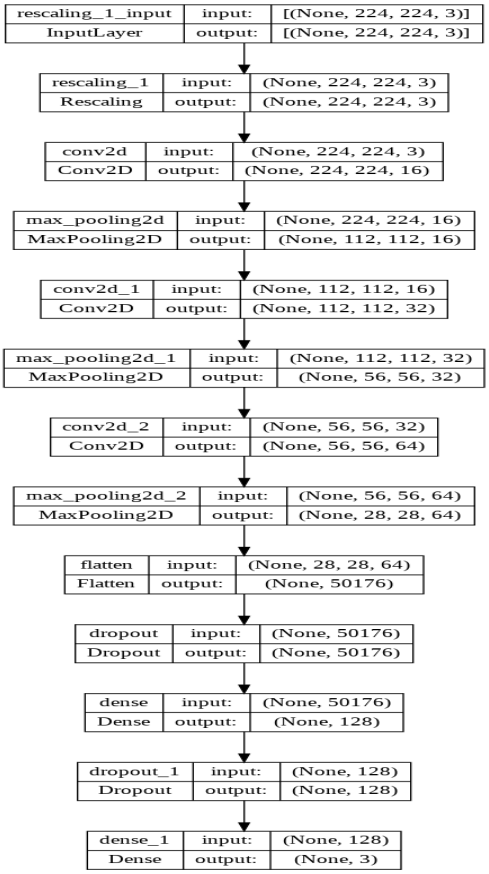

Figure 4. Customized CNN Model.

We first built a customized CNN classifier as shown in Figure 4. After multiplying the input image pixel by 1/255 to scale the values between 0 and 1, three 2D convolutional layers followed by max pooling layers are stacked. Finally, two dense layers are attached on top to output 3 numbers corresponding to three target classes. Dropout layers are inserted in between to avoid overfitting.

Transfer learning is a machine learning technique where a model trained on one task is used as a starting point for a model on a second, related task. In the context of Convolutional Neural Networks (CNNs), transfer learning is the process of using a pre-trained CNN as a starting point for training a new CNN on a different dataset. This is done by taking a pre-trained CNN, such as one trained on a large dataset like ImageNet, and using its learned feature maps as the starting point for training a new CNN on a different dataset. The new CNN can then be fine-tuned on the new dataset, allowing it to learn task-specific features while still leveraging the knowledge gained from the pre-trained model. This can significantly reduce the amount of data and computational resources needed to train a high-performing CNN on a new task.

TensorFlow Hub is a repository of trained machine learning models that can either be directly deployed or fine-tuned for customized tasks. There are hundreds of image classification models trained on various large datasets (e.g. Imagenet). Corresponding to each model, a feature extractor is also provided. When using a pre-trained feature extractor for transfer learning, the initial layers of the pre-trained CNN are used as a fixed feature extractor, and the final layers are replaced with new layers that are trained on the new dataset for the new task. This allows the model to leverage the knowledge gained from the pre-trained model, while still being able to adapt to the specific characteristics of the new dataset. In Figure 5, the highlighted layer indicates the instantiation of pre-trained feature extractors from Tensorflow Hub. It outputs a feature map of length 1280 in this case. Similar to the customized CNN model built earlier, two dense layers with one dropout layer in between are attached and will be trained to our own data for three formations identification.

Among hundreds of feature extractors available in TensorFlow Hub, we picked the following few representative architectures for comparison:

MobileNet: is a family of mobile-friendly CNN architectures that are designed to be light-weight, efficient, and perform well on mobile devices with limited computational resources. MobileNet models are built using depth-wise separable convolutions, which are a form of convolution that applies a single filter to each input channel, as opposed to the standard convolution that applies a different filter to each input channel. This allows for a reduction in the number of parameters and computation required, while still preserving the ability to extract features from the input.

Inception: The key feature of the Inception model is the use of multiple parallel convolutional filters of different sizes, which are applied to the input image at the same time. This allows the model to learn features at multiple scales, which can be useful for recognizing objects in images with varying sizes and orientations.

EfficientNet: The main goal of the EfficientNet architecture is to improve the efficiency of CNN models by automatically scaling up their capacity (i.e., the number of filters and layers) while also scaling up the input image resolution. The architecture also uses a compound scaling method that scales up the depth, width, and resolution of the model in a more optimal way. The EfficientNet architecture achieved state-of-the-art performance on image classification tasks while being more efficient in terms of computation and the number of parameters compared to previous models such as Inception and MobileNet.

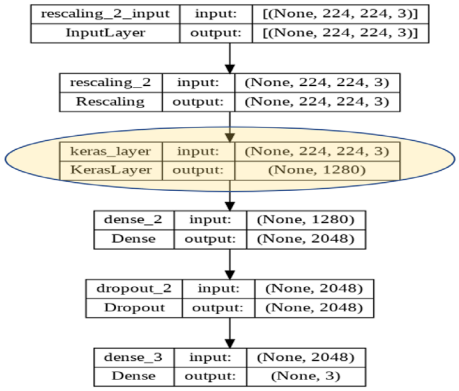

Figure 5. Transfer Learning Model using pre-trained feature extractor.

ResNet-50: The main idea behind the ResNet architecture is to use "shortcut connections" or "skip connections" that bypass one or more layers and allow the information to flow through the network more directly. This allows the network to learn the residual mapping, which can be easier for the network to learn compared to the original mapping. This idea of residual learning allows the network to learn more complex functions and improve the accuracy of the model. The original ResNet architecture had 152 layers and has since been extended and improved upon, with deeper and wider versions such as ResNet-50.

ViT (Vision Transformer): ViT (Vision Transformer) is a variant of the transformer architecture, which was originally designed for natural language processing tasks such as language translation. The main idea behind the ViT architecture is to treat images as a sequence of patches rather than a grid of pixels, and then apply transformer-based models to process this sequence of patches. The input image is divided into a fixed-size grid of non-overlapping patches and each patch is then flattened and fed into a transformer-based model. The transformer model learns the representations of the patches and the relationships between them.

Table 2 compares the model parameters of the customized CNN model and selected transfer learning models. While the custom CNN model has around 6.4 million parameters and all are trainable, the trainable parameters for the transfer learning models range from 794K to 4.2M. As mentioned earlier, MobileNet is lightweight with only 2.2 million pre-trained parameters compared to other models that all have more than 20 million pre-trained weights.

Table 2. Image Classification Models Parameter Comparison.

	Custom CNN	MobileNet (v2)	Inception (v3)	EfficientNet (v2)	Resnet50	ViT
Total Params	6446627	4887619	26005283	22960995	27702851	22460291
Trainable Params	6446627	2629635	4202499	2629635	4202499	794627
Non-trainable params	0	2257984	21802784	20331360	23500352	21665664

Results

Model Training

The Adam optimizer is used to train all the models for 40 epochs. The customized CNN model uses the default learning rate of 0.001 while all other transfer learning models use a smaller learning rate of 0.0005 due to the majority of the weights of the model being pre-trained.

Table 3 compares the training time per epoch using GPU on Google Colab. Both the customized CNN model and MobileNet only take 2 seconds due to their limited model capacity (6 to 7 million total parameters). Among all other models with a similar number of parameters (greater than 20 million total parameters), Resnet-50 is the slowest and Inception only takes 4 seconds to train an epoch even though it has ~26 million total parameters.

Table 3. Training Time Comparison.

	Custom CNN	MobileNet (v2)	Inception (v3)	EfficientNet (v2)	Resnet-50	ViT
GPU Training time (seconds per Epoch)	2	2	4	7	11	6

Training and Validation Accuracy/Loss Curve

Figure 6 shows both accuracy and loss curve for training and validation over the 40 epochs. Even though dropout is used in all the models to mitigate overfitting, validation accuracy flattened at various levels while training accuracy reaches 100% (except for Resnet-50) indicating some sort of overfitting that may be due to insufficient training data. One area for future improvement is to collect more video clips for each formation.

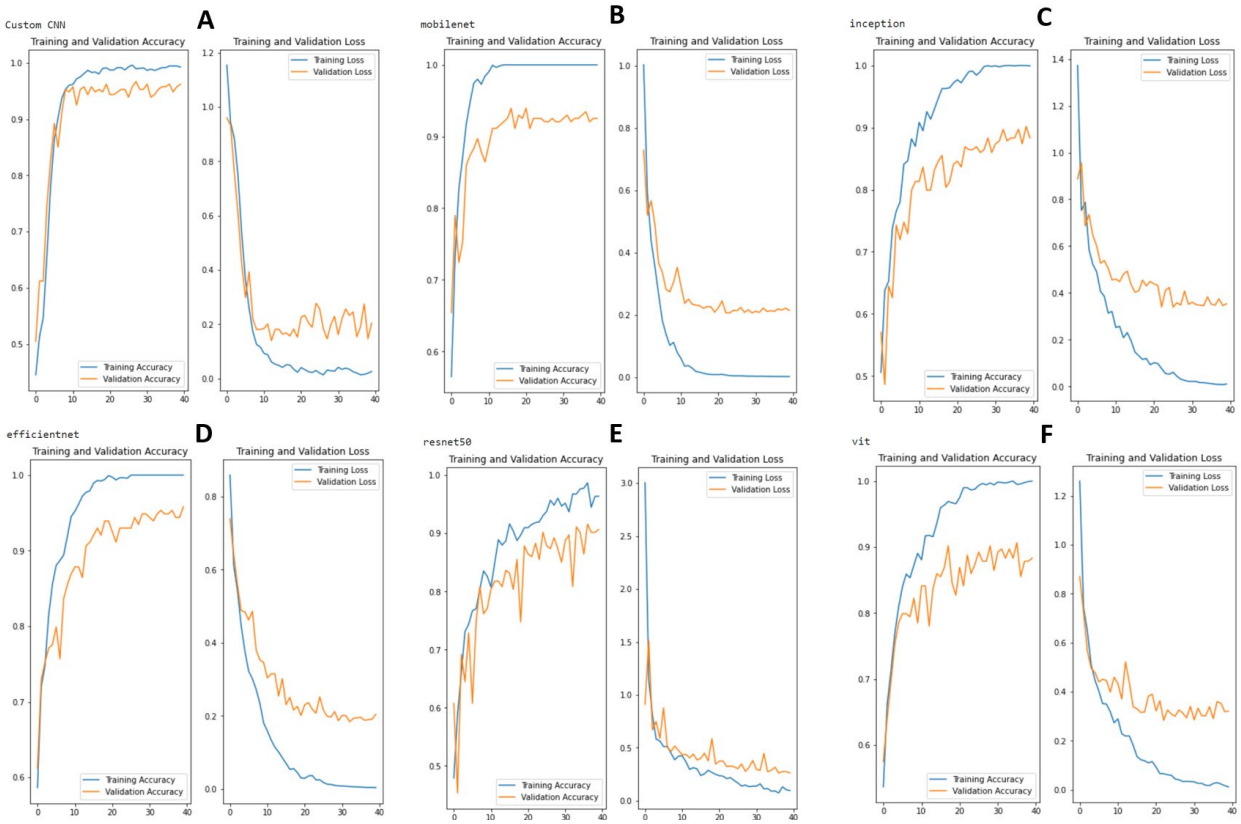

Figure 6. Model Training and Validation Accuracy/Loss Curve. A - Custom CNN; B – MobileNet; C – Inception; D – EfficientNet; E – ResNet50; F – Vision Transformer (ViT)

Test Accuracy

For each model, we use the reserved test dataset to report classification results including accuracy, precision, recall, and f1-scores. Figure 7 shows the classification report for the customized CNN model with overall 94% accuracy on the test dataset.

	precision	recall	f1-score	support
DoublesPictures	0.95	0.91	0.93	155
EmptyPictures	1.00	0.93	0.96	44
ReoPictures	0.92	0.96	0.94	185
accuracy			0.94	384
macro avg	0.95	0.93	0.94	384
weighted avg	0.94	0.94	0.94	384

Figure 7. Custom CNN Classification Report on Test Dataset

Figure 8 shows the comparison of model test accuracy. EfficientNet reaches the highest accuracy of 96% followed by customized CNN and MobileNet which achieve 94% and 93% accuracy respectively.

Recently, transformer-based architectures have gained significant popularity and have demonstrated state-of-the-art results in multiple domains beyond natural language processing, including computer vision. ViT (Vision Transformer) is a transformer-based architecture that has been shown to achieve state-of-the-art results on image classification tasks. One of the key characteristics of ViT is that it treats an image as a sequence of

tokens, rather than a 2D array of pixels, which allows it to take advantage of the powerful self-attention mechanism of transformers. However, this approach also means that ViT models typically require more data to perform well compared to other architectures such as CNNs. This is because the self-attention mechanism requires large amounts of context to work effectively, and this context is provided by the large amounts of data used during training. Additionally, since the tokens can be seen as patches of an image, the more patches, the more diverse and richer the representation, hence more data is needed to learn all the possible variations. This may be a contributing factor to the lower performance of the ViT model, as it only achieved a test accuracy of 89%.

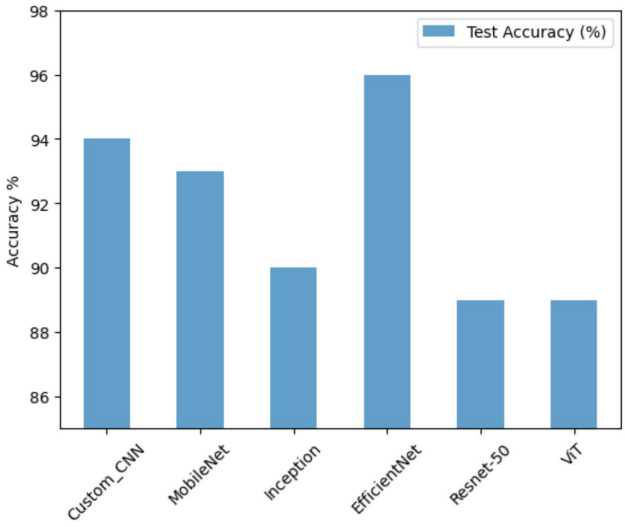

Figure 8. Model Test Accuracy Comparison.

Inference Results

In the process of inference, each video clip of a play goes through a preparatory step where several images from a play clip are selected and sent to a pre-trained model for identifying the formation of the play. The formation is established at the beginning of each play, but the starting point of the video can vary. To overcome this uncertainty, a time window of 2-4 seconds is chosen, and images are taken at a rate of half a second within the window. For instance, 8 frames can be taken from the start of the video, at a rate of half a second, which would cover the first 4 seconds of the video. Each of the sampled frames goes through the trained classification model and the majority vote decides the outcome.

We compared this multi-frame approach with only using one frame at a fixed time (the frame at the beginning of the second video). The study also looks at how the number of frames used, and the time window of sampling affect the results. In this experiment, the trained customized CNN model is used to analyze all 499 video clips, which belong to 3 different formations. The inference accuracy is measured as the number of correctly classified videos divided by the total number of video clips tested.

Figure 9-A presents different methods for sampling frames, each with varying start times and window sizes. For example, 4_frames_S_0 refers to sampling 4 frames starting at the beginning of the video, while 5_frames_S_1 indicates taking 5 frames starting at 1 second, with a sampling rate of half a second. The baseline method, 1_frame_fix, uses only 1 frame taken at the first second of the video. Figure 9-B compares the inference accuracy of these methods. The 5_frames_S_1 method achieves an accuracy of 95.4%, which is better than the baseline 1_frame_fix method. However, other multi-frame schemes perform worse than the baseline, likely due to the presence of noisy frames (players are still in motion and getting ready for the play) in the first second.

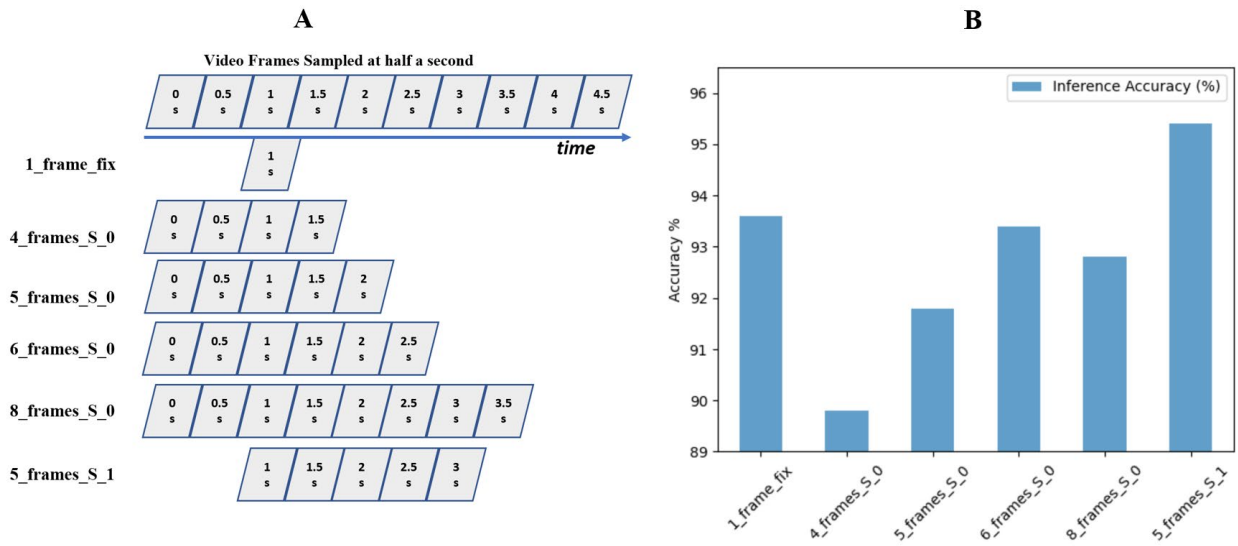

Figure 9. Inference Scheme and Accuracy Comparison. A – using one fixed frame vs. using multiple frames at different time windows. B – Video inference accuracy comparison.

Discussion

In our study, a batch size of 32 was used during the training of the image classification model. It would be worth exploring the impact of using other batch sizes, such as 16 and 64, on the model's performance. Another area for further research is to investigate other techniques for addressing overfitting, such as data augmentation and layer regularization. Additionally, it would be beneficial to evaluate more recent architectures that have been developed to improve the efficiency and accuracy of image classifiers. A few are listed below:

- ResNeSt (Zhang et al., 2020): This model is an extension of the ResNet architecture that incorporates a simple yet effective split-attention mechanism to improve performance.
- RegNet (Radosavovic et al., 2020): is a family of CNNs that are designed to be more scalable and efficient than previous models.

Recent years have seen significant progress in the field of video understanding, with several state-of-the-art models for video classification achieving high accuracy. The Two-Stream CNN (Simonyan et al., 2014) is a popular model that analyzes both visual and motion information by using spatial and temporal CNNs. The Temporal Segment Networks (TSN) (Wang et al., 2016) break down videos into segments for classification and then aggregate the results. 3D CNNs (Ji et al., 2013), which use 3D convolutional kernels like C3D, I3D, and SlowFast, have also gained popularity for their ability to learn both spatial and temporal features. TimeSformer (Bertasius et al., 2021), a transformer-based model, is also gaining attention for its ability to process video frames as a sequence of image patches and use self-attention mechanisms to learn spatiotemporal representations of the video.

It would be interesting to compare the performance of these video-based models to the image classifier evaluated in this paper. The formation of football teams at the beginning of each play may mean that including later video frames would not be beneficial. An important next step would be to develop methods for automatically identifying different offensive play types. Incorporating temporal information is likely to be crucial for this task.

Incorporating object detection techniques is a potential avenue for future research in the field of video understanding. Object detection can provide more detailed information about the objects and their locations

within the video, which can be used to improve the performance of the model. For example, by using object detection to identify and locate the players on a football field, it would be possible to gain insight into the formation of the team. However, it is important to keep in mind that using object detection techniques may come with added computational costs and may require a large amount of labeled data.

Conclusion

Machine learning is increasingly being used in sports analytics to gain insights and improve performance. It can be used to analyze large amounts of data, such as video footage, statistics, and sensor data, to identify patterns, predict outcomes, and make strategic decisions. In American football, studying the offensive formations used by opposing teams in past games is critical for developing a successful play strategy. This is because different formations can signal the type of plays that a team is likely to run, and by understanding these tendencies, coaches can anticipate and prepare for them. Manually labeling offensive formations in video clips can be a tedious and error-prone process. The process requires a person to watch the video clip and identify the formation used at each point in the game, which can be time-consuming and can lead to inconsistencies in the labels. Additionally, it is also prone to human error, as people may misidentify formations or make mistakes in the labeling process.

This paper presents a machine learning pipeline that can automatically label offensive formations in video clips. The pipeline includes video pre-processing, image classification, and a novel inference approach. The study compares a custom CNN model with pre-trained image classifier models using transfer learning, including architectures such as MobileNet, Inception, EfficientNet, and the transformer-based Vision Transformer (ViT). The models were trained on approximately 1400 images with the three most popular formation labels extracted from high-school football team matches. The results demonstrate that several models, including the custom CNN model, achieved over 90% classification accuracy on the test dataset. An inference is done by sampling multiple frames from a video clip at a fixed rate, passing them through a trained classifier, and taking a majority vote to determine the outcome. Our study found that the optimal inference accuracy of 95.4% is achieved by sampling five frames at a rate of 0.5 seconds starting at 1 second with a trained customized CNN model.

This machine learning system for identifying offensive formations in American football videos can assist high school and college football coaches in analyzing game footage. By automating the process of formation identification, the system can save coaches time and effort, allowing them to focus on other aspects of game analysis and strategy development.

Limitations

Getting labeled video clips that cover all the possible formations is a challenging task. In this study, we focused on the three most popular formations from our school football games. However, to make the model more generalizable, an ongoing effort is to collect more videos that cover a greater variety of football formations. This would allow us to train the models on a more diverse set of data, and ensure that the classification accuracy remains high even when the model is applied to new, unseen formations. Additionally, we also plan to evaluate the model's performance on a larger dataset of football games from different schools, colleges, or professional teams to increase the diversity of the data and evaluate the robustness of the model. Furthermore, we are looking into incorporating more advanced techniques and the latest image/video classification models to improve the performance and generalization of the model.

Acknowledgments

This research was mentored and supported by my computer science teacher Mr. Jason Galbraith as part of the Artificial Intelligence class I took this semester. I also want to thank the Sunset High School football program for providing me with labeled game film from the past season.

References

Ajmeri, O., & Shah, A. (2018). MIT Sloan Sports Analytics Conference. In Using Computer Vision and Machine Learning to Automatically Classify NFL Game Film and Develop a Player Tracking System. Boston. Retrieved December 31, 2022, from https://www.sloansportsconference.com/research-papers/using-computer-vision-and-machine-learning-to-automatically-classify-nfl-game-film-and-develop-a-player-tracking-system.

Atmosukarto, I., Ghanem, B., Ahuja, S., Muthuswamy, K., & Ahuja, N. (2013, June). Automatic Recognition of Offensive Team Formation in American Football Plays. In 2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops (pp. 991-998). https://doi.org/10.1109/CVPRW.2013.144

Bertasius, G., Wang, H., & Torresani, L. (2021). Is Space-Time Attention All You Need for Video Understanding? https://doi.org/10.48550/ARXIV.2102.05095

Dickmanns, L. (2021). Pose Estimation and Analysis for American Football Videos (dissertation).

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2020). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. arXiv. https://doi.org/10.48550/arxiv.2010.11929

Feichtenhofer, C. (2020). X3D: Expanding Architectures for Efficient Video Recognition. CoRR, abs/2004.04730. https://doi.org/10.48550/arXiv.2004.04730

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 770-778).

Hess, R., Fern, A., & Mortensen, E. (2007). Mixture-of-parts pictorial structures for objects with variable part sets. In Proceedings of the International Conference on Computer Vision (ICCV).

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications. https://doi.org/10.48550/arXiv.1704.04861

Ji, S., Xu, W., Yang, M., & Yu, K. (2013). 3D Convolutional Neural Networks for Human Action Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(1), 221-231. https://doi.org/10.1109/TPAMI.2012.59

Li, Z., Liu, F., Yang, W., Peng, S., & Zhou, J. (2022). A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects. IEEE Transactions on Neural Networks and Learning Systems, 33(12), 6999-7019. https://doi.org/10.1109/TNNLS.2021.3084827

Newman, J. D. (2022). Automated Pre-Play Analysis of American Football Formations Using Deep Learning. Theses and Dissertations, 9623.

Radosavovic, I., Kosaraju, R. P., Girshick, R., He, K., & Dollár, P. (2020). Designing Network Design Spaces. arXiv. https://doi.org/10.48550/arxiv.2003.13678

Ribani, R., & Marengoni, M. (2019, August). A Survey of Transfer Learning for Convolutional Neural Networks. In 2019 32nd SIBGRAPI Conference on Graphics, Patterns and Images Tutorials (SIBGRAPI-T) (pp. 47-57). https://doi.org/10.1109/SIBGRAPI-T.2019.00010

Simonyan, K., & Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. Advances in neural information processing systems, 5, 568-576.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2015). Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1-9).

Tan, M., Le, Q. V., & Doraswamy, P. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. https://doi.org/10.48550/arXiv.1905.11946

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In Advances in neural information processing systems (pp. 5998-6008).

Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., & Van Gool, L. (2016). Temporal Segment Networks: Towards Good Practices for Deep Action Recognition. arXiv. https://doi.org/10.48550/ARXIV.1608.00859

Zhang, H., Wu, C., Zhang, Z., Zhu, Y., Zhang, Z., Lin, H., Sun, Y., He, T., Mueller, J., Manmatha, R., Li, M., & Smola, A. (2020). ResNeSt: Split-Attention Networks. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2735-2745.