Exploring Computer Vision Tools for Division III Football

Thomas McConnell
Pomona College
Claremont, CA
tmma2020@mymail.pomona.edu

1. Motivation

The world of sports analytics is flush with data-driven insights, in part due to the overall increase in data quality and quantity, as well as the expansion of cheap compute access. While the NFL and high-level NCAA American football programs, teams with substantial resources have begun utilizing computer vision [6], teams like Pomona-Pitzer or Claremont-Mudd-Scripps have yet to implement a computer vision tool to assist in operations. Although a tool like Amazon Prime's blitz predictor [6] may be unrealistic, even a tool that labels plays and formations would save lower-budget programs substantial amounts of time.

A tool like Amazon's was unrealistic primarily because DIII data collection does not include the ground truth labeling that would enable such a tool. For example, DIII data collection does not label field lines, so even identifying the field for a specific field/camera setup would take substantial time to achieve sufficient accuracy to replace a human.

As a proud Pomona-Pitzer football alumnus, I thought a project that attempted to complete some of the tasks coaches normally do by hand would be a stimulating way to gain experience with computer vision. Given the relative popularity of DIII football in comparison to the NFL and better college teams, the body of academic literature specifically relating to DIII film was limited.

However, papers like Chen *et al*. [2] took steps to qualify and work within the constraints of lower-level football program data. Specifically, some of the challenges of film data were: inconsistent video quality, filmed on different cameras, often with motion blur; inconsistent field painting, making field identification challenging; inconsistent plays, where plays were sometimes missing from the overall data; and inconsistent angles, where one game's film might include five different camera angles, and another might only include one or two.

Figure 1. Example of sideline view football film used in project. Note how hard it is to easily see the white lines on some parts of the field and to distinguish between some players.

Looking at Figure 1, we can see how hard it would be to create a useful computer vision tool with such inconsistent data. As such, my project aimed to combine multiple existing computer vision techniques to identify players and formations despite these shortcomings of the data.

2. Background

2.1. YOLOv8x

YOLOv8x is the latest iteration of the You Only Look Once (YOLO) object detection models, developed by Ultralytics [7]. It offers improved accuracy, speed, and efficiency over its predecessors. YOLOv8x is particularly well-suited for real-time tracking of small, fast-moving objects, making it ideal for football film analysis where players and the ball must be detected across varying camera angles and resolutions. In this project, I used the YOLOv8x model, the largest available version, to maximize detection accuracy given the challenging conditions of Division III football film.

2.2. ResNet-18

ResNet-18 is a deep convolutional neural network (CNN) architecture consisting of 18 layers, introduced by He *et al*. [3]. It is widely used for image classification and feature extraction tasks. I chose ResNet-18 for this

project because of its strong performance on image recognition benchmarks combined with its relative computational efficiency compared to deeper architectures. This balance made it well-suited for classifying football formations from both raw and processed images, where computational resources and training time were considerations.

2.3. Random Forest

Random Forest is an ensemble machine learning method introduced by Breiman [1]. It constructs multiple decision trees during training and outputs the mode of the classes for classification tasks. I used Random Forest to classify offensive football formations based on spatial distributions of detected players after YOLO-based object detection. Its robustness to noise and strong performance on structured tabular data made it an appropriate choice for this classification task.

3. Methodology

My project aimed to build a Division III-appropriate player and formation identification pipeline, inspired by the approach of Newman *et al*. [5], which used bounding box detection and spatial distribution analysis to classify football formations. To adapt this for DIII film constraints, I first applied a series of preprocessing steps to sideline images to isolate the field by filtering out non-green regions, detect field lines, and crop images to the primary field of play. These cropped images were then passed through a YOLOv8x model to detect players and output bounding boxes representing their locations.

The centroid coordinates of detected players were extracted and used as input features for a Random Forest classifier to predict the offensive formation. This pipeline—cropping, YOLO detection, and Random Forest classification—served as the core focus of my project.

To establish comparative baselines, I also evaluated:

- Performance without any preprocessing (using full raw images).
- A ResNet-18 model trained directly on raw images for formation classification.
- A ResNet-18 model trained on preprocessed (cropped) images for formation classification.

This comparative analysis allowed me to assess the impact of preprocessing and different modeling strategies on formation identification accuracy.

3.1. Outline

The steps of my project were:

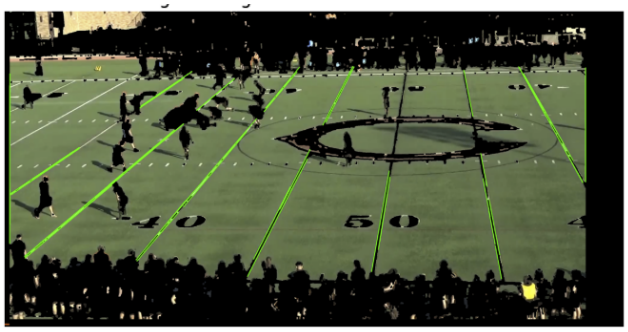
1. Collect Pomona Pitzer film data from Hudl, the video processing application used by most DIII programs. Data collected included: Presnap images from all available camera angles for a given play (sidelines, endzone, tight), play metadata (formation, down and distance, play result, play type)
2. Preprocess sideline images by applying field masking and cropping techniques.
3. Detect players in the preprocessed images using YOLOv8x to generate bounding boxes.
4. Extract centroid coordinates from the bounding boxes.
5. Classify offensive formations using a Random Forest model based on player centroids.
6. Compare model performance across:
   (a) Cropped images vs. raw images.
   (b) Random Forest pipeline vs. ResNet-18 classification models.

4. Results

4.1. Data Description

The data used in this project consisted of Pomona-Pitzer Hudl film, which included both game and practice recordings. Each play was labeled by coaches with information such as offensive formation, play type, and game situation. I extracted a presnap still frame from each play for use in classification tasks. These frames were used as inputs for both the object detection and classification pipelines described in Section 3. My data ended up having 1098 photos, and I used and 80 / 20 split for testing and training each model.

4.2. Processing Pipeline Demonstration



Field Detection Mask

Final Cropped and Color-Corrected Image

Figure 2. Processing pipeline steps: field mask generation (left) and final cropped field image (right) used for player detection.

Before analyzing model performance, Figure 2 demonstrates the preprocessing steps applied to raw Hudl film footage. First, the field was detected by masking for only green colors. Then, field lines were detected to generate a mask of the field boundaries, as shown in the top image. I used the most conservative line (the highest line that was detected and was within 10 degrees of being vertical) to determine the top of the field. I did the same to determine the bottom of the field. Then, the field area was cropped to produce the final input image for the YOLOv8x model, as shown in the bottom image.

4.3. Model Performance

I evaluated four approaches to formation classification:

1. **YOLOv8x + Random Forest (Processed Images)**: Player bounding boxes were extracted from cropped images, and centroids were used as features for Random Forest classification.
2. **YOLOv8x + Random Forest (Raw Images)**: Player bounding boxes were extracted directly from raw images without preprocessing, then classified with Random Forest.
3. **ResNet-18 (Processed Images)**: Formation classification was performed end-to-end using ResNet-18 trained directly on cropped (processed) images
4. **ResNet-18 (Raw Images)**: Formation classification was performed end-to-end using ResNet-18 trained directly on raw presnap images.

4.4. Evaluation Summary

The primary evaluation metric was formation classification accuracy, measured on a held-out validation set.

4.5. YOLO-Based Model Results

	precision	recall	f1-score	support
Aces	0.18	0.20	0.19	30
JOKERS	0.22	0.12	0.16	33
KINGSSPLIT	0.32	0.76	0.45	21
Kings	0.21	0.14	0.17	21
LIGHTNING	0.09	0.08	0.08	26
MIAMI	0.15	0.12	0.13	26
PRO	0.33	0.42	0.37	31
QUEENS	0.23	0.16	0.19	32
accuracy			0.24	220
macro avg	0.22	0.25	0.22	220
weighted avg	0.22	0.24	0.21	220

(a) YOLOv8x detection after preprocessing and cropping

	precision	recall	f1-score	support
Aces	0.25	0.30	0.27	30
JOKERS	0.17	0.12	0.14	33
KINGSSPLIT	0.23	0.33	0.27	21
Kings	0.33	0.19	0.24	21
LIGHTNING	0.31	0.19	0.24	26
MIAMI	0.00	0.00	0.00	26
PRO	0.26	0.39	0.31	31
QUEENS	0.18	0.19	0.18	32
accuracy			0.21	220
macro avg	0.22	0.21	0.21	220
weighted avg	0.21	0.21	0.21	220

(b) YOLOv8x detection on raw image (no preprocessing)

Figure 3. Comparison of formation detection between raw and preprocessed frames utlizing YOLO8x and a Random Forest classifier.

Looking at Figure 3, the YOLO-based models, which relied on detecting player bounding boxes and feeding their centroid coordinates into a Random Forest classifier, showed poor overall performance. Both the raw image pipeline and the processed (cropped) image pipeline struggled to produce useful classification outcomes.

The processed images YOLO-based model achieved an accuracy of approximately 22% macro average, while the raw version also achieved only 22% macro average. As such, the preprocessing steps provided no significant benefit to recognizing formations. The low accuracy suggests that YOLOv8x struggled to consistently and accurately detect

all players in each frame, introducing noise that critically degraded formation prediction performance.

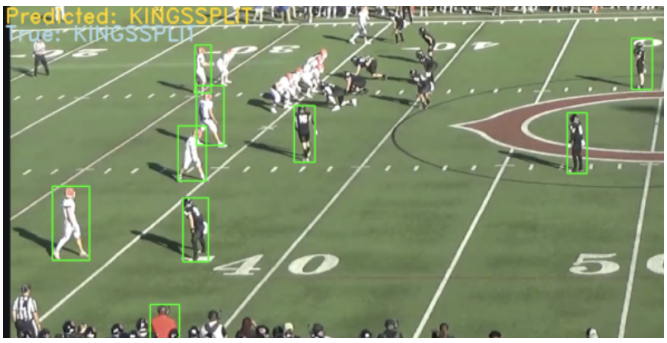


Figure 4. Example of yolo player detection and the resulting prediction.

Looking at Figure 4, we can see that while the prediction was accurate, the actual players detected is both incomplete and inaccurate. We errantly detect a coach on the sideline who is not part of the player, and we fail to detect many of the interior players. This demonstrates how the YOLO model might have struggled to detect players, which would've made the subsequent prediction challenging.

4.6. Comparison to ResNet-18

	precision	recall	f1-score	support
Aces	0.21	0.18	0.19	28
JOKERS	0.61	0.39	0.47	36
KINGSSPLIT	0.36	0.97	0.53	29
Kings	0.33	0.05	0.09	19
LIGHTNING	0.29	0.25	0.27	24
MIAMI	0.29	0.59	0.39	29
PRO	0.57	0.12	0.21	32
QUEENS	0.83	0.22	0.34	23
accuracy			0.36	220
macro avg	0.44	0.35	0.31	220
weighted avg	0.44	0.36	0.33	220

(a) ResNet-18 detection after preprocessing and cropping

	precision	recall	f1-score	support
Aces	0.56	0.33	0.42	27
JOKERS	0.48	0.48	0.48	31
KINGSSPLIT	0.83	0.47	0.60	32
Kings	0.40	0.52	0.45	27
LIGHTNING	0.38	0.60	0.46	20
MIAMI	0.53	0.57	0.55	30
PRO	0.79	0.52	0.62	29
QUEENS	0.43	0.67	0.52	24
accuracy			0.51	220
macro avg	0.55	0.52	0.51	220
weighted avg	0.57	0.51	0.52	220

(b) ResNet-18 detection on raw image (no preprocessing)

Figure 5. Comparison of formation detection between raw and preprocessed frames utilizing ResNet-18.

Looking at Figure 5 The ResNet-18 models, which directly classified formations from full images without intermediate player detection, performed significantly better than the YOLO-based pipelines. The raw images actually helped the model perform better than the preprocessed images, as the macro average for the model that used raw images as input was 55%, while the macro average for the model that used the preprocessed images was only 44%. Regardless, both models' accuracy was at least double that of the models utilizing YOLO.

By skipping object detection altogether and allowing the ResNet model to work directly on the full image, stronger spatial feature learning was achieved.

However, even the ResNet models achieved accuracy below the near-perfect accuracy needed for practical deployment in a coaching environment. Thus, the proposed pipeline, in its current form, did not yield a tool capable of replacing human analysts for football film annotation tasks.

5. Discussion and Limitations

5.1. Challenges of Applying Computer Vision to DIII Football Film

As discussed in the motivation, there exists a dearth of research and tools for computer vision in Division III (DIII) sports, especially football. Without an extensive body of prior literature, there were fewer established best practices or benchmarks to guide project development. This lack of direction increased the risk of methodological dead-ends.

5.2. Difficulties in Player Detection and Field Inconsistencies

One of the clearest findings from this project was the difficulty in reliably detecting all players on the field across raw DIII film. Johnson *et al*. [4] previously noted the challenges of consistent player localization even with higher quality NFL data. My project reinforced this issue: despite preprocessing steps like field masking and cropping, YOLOv8x frequently failed to consistently detect all players, degrading the downstream classification task. Furthermore, differences between camera setups, and inconsistent field painting severely impacted model generalization. While image preprocessing steps we supposed to improve model performance, we can see that model performance did not notably improve with processing, and actually decreased with the ResNet models. Even with supposed improved input quality, object detection models struggled with player occlusions, resolution loss, and misidentifications—problems in sports vision literature.

5.3. Limitations in Training Data and Labeling

This project was constrained by limited access to diverse training data. Only Pomona-Pitzer game film was available, introducing potential biases. A larger and more standardized dataset could have improved model robustness. Additionally, significant labeling of the existing training data for things such as field lines, player locations, ball location, and the line of scrimmage might have enabled much more robust training. But given that the goal of the project was to create a tool that used the existing data, this data was unavailable for training.

5.4. Comparison of YOLO and ResNet Pipelines

The ResNet-18 models, which skipped intermediate player detection and operated directly on raw images, substantially outperformed the YOLO-based pipelines. This suggests that for DIII data, direct image-based classification may be a more promising approach than trying to first localize individual players. However, even ResNet-18 only achieved 55% accuracy—still well below practical requirements for real-world use in coaching workflows. In addition, the computational requirements for the ResNet model training far exceeded those that a normal DIII football program would have access to. As discussed, one goal of the project was to create a budget friendly tool. However, each team would realistically need to train their own model for a tool such as this because each team would have their own formation data. Thus, using the teapot server, with its two relatively expensive and powerful GPUs, is not a realistic ask for most teams. The ResNet models each took about 20 minutes to train, indicating that the overall compute needed might exceed what a team has access to.

5.5. Overall Takeaways

The results demonstrate that building a computer vision tool for DIII football film is significantly more difficult than for professional or high-level college data, primarily due to film inconsistencies and labeling limitations. Despite careful preprocessing and model selection, the final pipeline did not achieve the accuracy necessary to replace human analysts, highlighting the challenges inherent to applying computer vision in this domain. Regardless, this project served as a great introduction to computer vision and the application of techniques learned in class. Additionally, the problems encountered were good learning experiences in problem solving as well as practical application of computer vision processes.

6. Access to Project Code

6.1. Github

The code for this project can be accessed at the Github repository found here. The repository contains sample notebooks and data, demonstrating the pipeline process used to create the tool. The repository does not contain the original data used, as it is not publicly available. The repository also contains the python scripts used to train the models on the teapot server. The Github repository clearly explains the necessary requirements and steps to replicate the work done to create this tool.

References

[1] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. 2

[2] Sheng Chen, Zhongyuan Feng, Qingkai Lu, Behrooz Mahasseni, Trevor Fiez, Alan Fern, and Sinisa Todorovic. Play type recognition in real-world football video. In *IEEE Winter Conference on Applications of Computer Vision*, pages 652–659, 2014. 1

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 770–778, 2016. 1

[4] Neil Johnson. Extracting player tracking data from video using non-stationary cameras and a combination of computer

vision techniques. In *MIT Sloan Sports Analytics Conference*, 2025. 5

[5] Jacob Newman, Andrew Sumsion, Shad Torrie, and Dah-Jye Lee. Automated pre-play analysis of american football formations using deep learning. *Electronics*, 12(3), 2023. 2

[6] The Athletic Staff. Thursday night football: A look at amazon prime vision's advanced analytics, October 2023. Accessed: 2025-03-08. 1

[7] Ultralytics. Yolov8: Open-source yolo models for object detection, segmentation, and classification, 2023. Accessed: 2025-05-05. 1