Using Computer Vision and Machine Learning to Automatically Classify NFL Game Film and Develop a Player Tracking System

Omar Ajmeri   Ali Shah
omarajmeri@gmail.com   ali@alternative.ly

Paper Track: Other Sports
Paper ID: #5571

1. Introduction

NFL coaches spend countless hours tagging and mining game film, searching for tendencies and patterns to exploit in upcoming matchups. Film tagging alone is a tedious and error-prone task – coaches need to label formations, personnel, and ideally, routes. Depending on the play, there can be up to 6 designed routes to account for, generating a wealth of data for each play. While self-scouting is fairly simple given a coach's understanding of his own playbook, competitively scouting every team in the league on a week to week basis is an extremely time-consuming task.

Through a series of computer vision techniques looking at pixel density and weighted spatial reasoning, we have automated the classification of NFL "All-22" game film from start (offensive formation labeling) to finish (video player tracking coordinates throughout the life of a play). This not only includes formations, but also player routes and player speeds. This effective player tracking system has implications for game planning, scouting, and better evaluation of individual players and coaches. The ability to analyze player location data on a mass scale in a short period of time will fundamentally change how football coaches scout and analyze players and opposing coaches throughout the league.

2. Data

Currently, NFL teams can only access the Next Gen Stats player tracking data for their own team, but can't access competitive team data. Thus, we decided to build our own algorithm that can capture player tracking data for all teams.

Without access to labeled NFL game film, the dataset needed to be developed manually. We chose the Washington Redskins home games as a test case, and captured screenshots at a rate of 5 frames per second.

Figure 1. *Example screenshot of NFL "All-22" game film*

After capturing the screenshots, there is a series of transformations that needs to occur to standardize the analysis. Because the film is shot at different angles, the first step is to automatically locate the field lines on the image and rotate the image accordingly to ensure that they are arranged perpendicularly. We tried a number of approaches here but ultimately decided on using the Hough Lines computer vision technique. Many of the lines we found were often superfluous, therefore we applied some further heuristics on these lines to leave only the full field white lines. Once we established the line of scrimmage based on the proximity to the offensive line, we then calculated the rotation of the camera relative to the field. This is achieved through the arccos(x) using the line of scrimmage and the accompanying perpendicular line. Extraneous parts of the image are removed as well, most notably the NFL logo in the bottom right hand corner.

Figure 2. *Detection of the line of scrimmage and rotation angle*

3. Formation Identification

Offensive players are identified algorithmically through a series of steps involving identifying the line of scrimmage and color analysis based on jersey color (burgundy in this case). The X and Y player coordinates are recorded pre-snap in relation to the quarterback, denoted through a blue square. After additional tweaks to the algorithm to account for the closely clustered offensive linemen, we created a training dataset of labeled images. This was done through automatically tagging over 500 images of formations with the coordinate locations of the players.
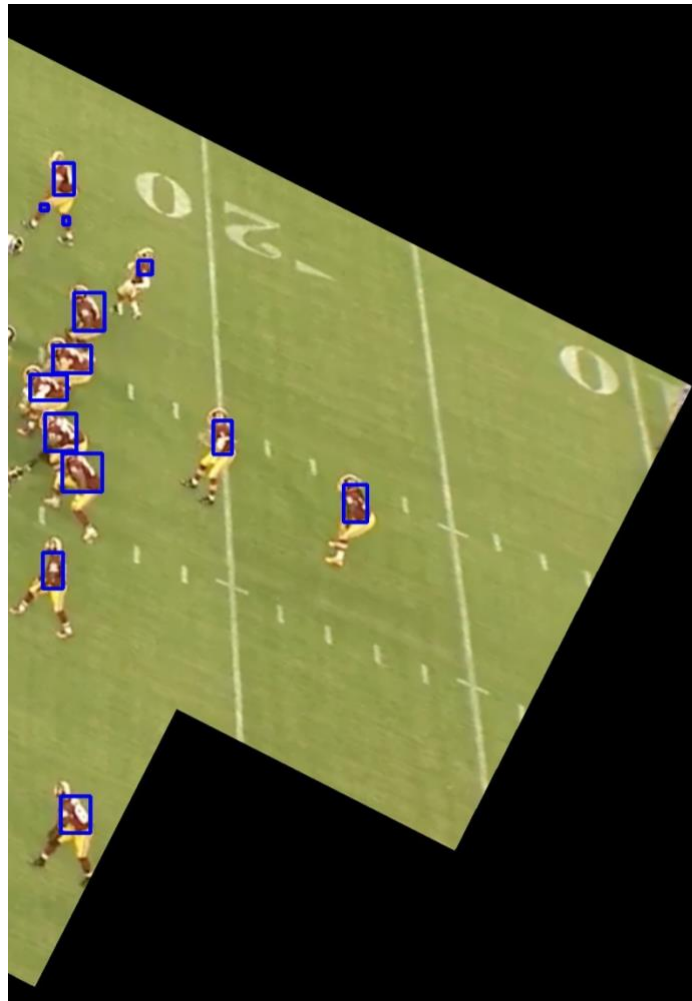

Figure 3. *Rotated image with detected players*

3.1 Player Locations

Multiple machine learning models were used to classify the formation. We tested 5 unique models using coordinate locations to first determine offensive personnel. Given the potentially vast differences between images with the same formation, models were run independently to assess player locations. Outcomes include quarterback position (Shotgun, Under Center, Pistol), as well as the number and location of running backs, receivers, and tight ends. Below is an example of two

formations that would be classified in the same way, but have a significant difference in player location.

Figure 4A-B. *Example of formations that look different but are classified in the same way*

After each of the features are determined independently, the formation is derived through the sum of the parts. Below is an accuracy chart for one of the features, quarterback position, to understand model performance overall.

Classifier	Accuracy %
Classification and Regression Trees	86.5%
Naïve Bayes	67.5%
SVM	56.1%
K Neighbors	49.8%
Logistic Regression	42.9%

Table 1. *Machine learning algorithm accuracy for QB Position*

Focusing on the best classifier, Classification and Regression Trees, the Classification Report provides interesting insights. The toughest quarterback position to correctly identify was Pistol. This intuitively makes sense, as the lack of consistent use of the formation limited our training set.

QB Position	Precision	Recall
Center	.82	.92
Shotgun	.90	.84
Pistol	.50	.12
Average	**.84**	**.85**

Table 2. *Classification report for QB position*

3.2 Formation Prediction

In addition to the machine learning algorithms used to identify player locations and groupings based on coordinate features, the original labeled dataset was used to train another model. After generating the coordinate locations and developing an algorithm for formation identification, the models were then combined to recognize formations (29 total) based simply on coordinate locations.

Classifier	Accuracy %
Classification and Regression Trees	72.3%
Naïve Bayes	68.8%
SVM	64.6%
K Neighbors	64.1%
Logistic Regression	55.6%

Table 3. *Machine learning algorithm accuracy for formation prediction*

Formation	Precision	Recall
Singleback Ace	.86	.90
Singleback Ace Pair Slot	.84	.74
Spread Center	.81	.88
Spread Gun	.81	.86
Empty Trips Gun	.76	.70

Table 4. *Classification report for formation identification (top 5 formations in terms of sample)*

Based on the results, the Classification and Regression Trees model proved most effective once again, predicting the correct formation about 72% of the time. With additional training data, we expect model performance to continue to increase over time. While this is simply a proof of concept, the model can continue to be tweaked for a variety of different applications including other NFL teams, NCAA teams, and even High School football teams. Coaches will save significant amounts of time as a result of this automatic classification system, with the ability to scout individual formations at scale in a much quicker way, as well as increase the capacity to do more macro and micro level analysis and scouting.

3.2 Play-by-play data
A powerful application of automatically categorizing formations lies within merging play-by-play data. By connecting the two data sources (as seen below), tendencies can be uncovered based on the outcome of the play by each formation.

Image	Formation	Play-by-play
STLatWAS_1	Singleback Ace	1ST & 10 AT WAS 19(13:45) (13:45) M.Jones right guard to WAS 17 for -2 yards (M.Brockers).
STLatWAS_2	Spread Gun	2ND & 12 AT WAS 17(13:12) (13:12) (Shotgun) K.Cousins pass short left to P.Garçon to WAS 21 for 4 yards (J.Jenkins).

Table 5. *Image name, formation, and NFL Gamepass play-by-play description*

The play-by-play details found within the NFL Gamepass UI can be broken out into several layers for additional analysis as seen below. To simplify the line of scrimmage yard line, it is standardized

from 1-99 (if the Redskins have the ball on the opponent's 20 yard line, the line of scrimmage would be 80).

Image	Formation	Down	Distance	LOS	Time in Quarter	Type	Direction	Run Loc	Pass Type	Complete	Intended Receiver	Yards Gained
STLatWAS_1	Singleback Ace	1ST	10	19	13:45	Run	Right	Guard	-	N/A	None	-2
STLatWAS_2	Spread Gun	2ND	12	17	13:12	Pass	Left	-	Short	Complete	P. Garçon	4

Table 6. *Detailed break-out of NFL Gamepass information*

The wealth of data accumulated for each play combined with the automatically classified formation data allows us to gain insights into coaching tendencies. Below is a sample analysis of the data with a few filters. We chose to analyze games from 2015, when highly regarded offensive mind Sean McVay called the plays. Other filters applied are 1st down and at least 6 yards to go, on the Redskins' own half, and with over 2 minutes left in the quarter. Overall, McVay was ~60% more likely to call a run play in this situation, preferring the left side (equally split on running left end vs left guard). Running backs Matt Jones and Alfred Morris combined for a yards per carry (YPC) of just 2.9 on these runs. The most common formation, Singleback Ace Pair Slot (two tight ends on the right side of the line, two wide receivers to the left of the line, with one running back), is analyzed below. The sample isn't especially high, however this is only looking at a subset of all 2015 data with very specific filters applied - expanding these would undoubtedly generate more powerful conclusions.

Singleback Ace Pair Slot Analysis

- Equally likely to pass or run out of this formation
- 65% of runs were to the right side, equally split between right guard and right end, with few runs up the middle
- 81% of passes were on short routes
  - Pierre Garçon and Matt Jones were the most common short route targets
  - Jordan Reed was the most common deep target
- Passes gained an average of 4.6 yards per play, while runs averaged just 2.5 yards per play

This analysis simply scratches the surface of what is possible. Formation and play-by-play data can be merged over multiple seasons (with the same play-caller) to rigorously analyze tendencies through data. Armed with this information, defensive coordinators and players will be better able to gameplan against opponents.

4. Player Tracking

As mentioned earlier, player locations are captured at a rate of 5 frames per second. As players travel throughout the life of a play, their X and Y coordinates are tracked continuously in two dimensions. Based on their original starting point, we can evaluate the distance traveled for each player through Euclidean distance.
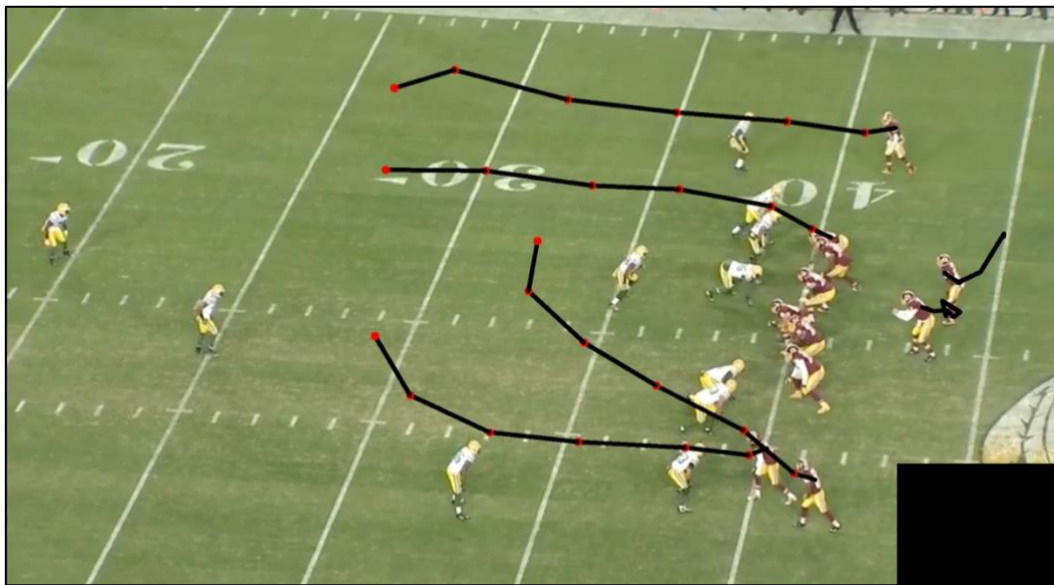
$$d(p,q) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2}$$

Figure 5. *Example of video player tracking routes and coordinates*

Once route distance is established, player speed can be uncovered based on the frame rate and distance traveled using the formulas below. Coordinate distance needs to be translated into a measureable unit, which in this case would be yards. Because there can be some variance with video player tracking data in the event that the camera moves, the distance is standardized through two tactics. First, the coordinate distance of 1 yard is derived individually for each screenshot using the distance between the full field white lines every 5 yards. Second, the coordinate location of the highest point in a full field white line in the first image is saved as a reference point, and is used to ensure that coordinate distances of players are not being overrepresented or underrepresented as the camera follows the play. Below is an example of DeSean Jackson's top speed in the above play.

$$\frac{215.5 \; distance \; (d)}{.2 \; seconds} \times \frac{1 \; yard}{\sim 82 \; d} = \frac{12.8 \; yards}{1 \; second}$$

After distance is converted into yards/second, a series of transformations then occurs to convert this into miles per hour.

$$\frac{12.8 \; yards}{1 \; second} \times \frac{3 \; feet}{1 \; yard} = \frac{38.4 \; feet}{1 \; second} \times \frac{1 \; mile}{5{,}280 \; feet} = \frac{.00728 \; miles}{1 \; second} \times \frac{60 \; seconds}{1 \; minute} \times \frac{60 \; minutes}{1 \; hour}$$

$$= 26.2 \; mph$$

This data can be used in a multitude of ways. While the speed appears slightly higher than we'd expect looking at Next Gen Stats, the speed numbers will only be compared relative to the speeds of other players. Looking at the first .6 seconds of the routes run by DeSean Jackson and Maurice Harris, we can better understand player acceleration and how fast these wide receivers are getting

off of the line of scrimmage. Based on the above formula and the coordinate locations of each, Jackson accelerates to 22.4 mph, while Harris only reaches 19.3 mph. This analysis can be replicated for all players, potentially in real time as images are delivered to Microsoft Surface tablets on NFL sidelines. Looking at average speed and acceleration throughout the course of a game can provide deeper insight into the rate of player fatigue, which coaches can then develop game plans around.

Personnel executives will also have a much clearer understanding of how effective players are at running different routes. For decades, coaches, analysts, and fans have had trouble reconciling the relationship between speed and route running. If Darrius Heyward-Bey runs a 4.3 second 40 yard dash, why can't he become one of the greatest wide receivers to play the game? One of the answers lies in route running, which until now, was mostly unquantifiable. While Heyward-Bey may have exceptional straight-line speed, coaches will now have the ability to compare how fast he runs a 10 yard dig route versus another receiver. This will be highly useful information for decision makers in terms of player scouting, determining free-agent signings, and contract negotiations.

Below is an example of two sets of highly similar routes run. In this "Stick" play concept, DeSean Jackson and Pierre Garçon each run "Comeback" routes, while Vernon Davis and Jordan Reed each run "Flat" routes. Using this player tracking algorithm, we are able to compare how effective these players are against the other.
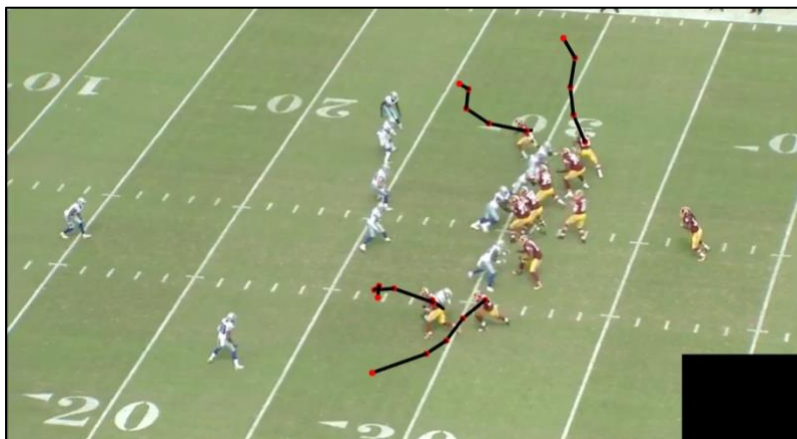

Figure 6. *Example of nearly identical routes with video player tracking*

To the naked eye in real-time, both sets of routes appear virtually identical - taking a closer look at the distance traveled and average speed confirms this for the "Flat" routes, as Davis and Reed almost exactly mirror each other. However, the wide receivers are a different story. Garçon runs a slightly more precise route and gets out of his break quicker than Jackson, as he performs almost a 180 degree turn compared to his counterpart's wide turn. As a result, Garçon runs 1.3 less yards than Jackson, and subsequently hauls in the pass from Cousins as he makes his move.

5. Conclusion

5.1 Summary

The automatic formation classification process has the potential to save coaches hours of time categorizing plays, not just for their own team, but also for competitive teams around the league. The additional layer of player tracking has enormous potential in its own right, with the ability to measure new metrics such as distance traveled and speed in real-time. As a result, coaches will be able to work both faster and smarter with these tools.

5.2 Limitations

With the limited number of training sample screenshots, we expect accuracy to increase with additional data. Because the algorithm uses jersey color to track players, there are some potential obstacles such as on-field shadows and sunlight affecting the RGB color. Gaining access to RFID player tracking data will allow this work to be continued to be built upon.

5.3 Future Work

While there are numerous applications to this work, we are simply scratching the surface with data collection and analysis. A major extension of this work would be replicating the analysis for defensive formations and player analysis. This would not only allow for the same types of analyses mentioned earlier albeit for defenses, but also the ability to examine the relationship between offensive and defensive play calling. Other aspects of the game to look into further include kickoff and punt coverage.

As stated earlier, this work also has broad applicability for all levels of football. As long as games are recorded using a stationary camera, the analysis can be replicated for the NCAA and even for High School football. The relatively low processing and time costs are bonuses as well – labeling formations for an entire game (~50 offensive plays) takes less than 5 minutes to complete, while player tracking expectedly tacks on additional time.