

Article

Automated Pre-Play Analysis of American Football Formations Using Deep Learning

Jacob Newman, Andrew Sumsion , Shad Torrie  and Dah-Jye Lee * 

Department of Electrical and Computer Engineering, Brigham Young University, Provo, UT 84602, USA

* Correspondence: djlee@byu.edu

Abstract: Annotation and analysis of sports videos is a time-consuming task that, once automated, will provide benefits to coaches, players, and spectators. American football, as the most watched sport in the United States, could especially benefit from this automation. Manual annotation and analysis of recorded videos of American football games is an inefficient and tedious process. Currently, most college football programs focus on annotating offensive formations to help them develop game plans for their upcoming games. As a first step to further research for this unique application, we use computer vision and deep learning to analyze an overhead image of a football play immediately before the play begins. This analysis consists of locating individual football players and labeling their position or roles, as well as identifying the formation of the offensive team. We obtain greater than 90% accuracy on both player detection and labeling, and 84.8% accuracy on formation identification. These results prove the feasibility of building a complete American football strategy analysis system using artificial intelligence. Collecting a larger dataset in real-world situations will enable further improvements. This would likewise enable American football teams to analyze game footage quickly.

Keywords: computer vision; deep learning; machine learning; American football; formation analysis



Citation: Newman, J.; Sumsion, A.; Torrie, S.; Lee, D.-J. Automated Pre-Play Analysis of American Football Formations Using Deep Learning. *Electronics* **2023**, *12*, 726. <https://doi.org/10.3390/electronics12030726>

Academic Editor: Chunjie Zhang

Received: 14 January 2023

Revised: 29 January 2023

Accepted: 31 January 2023

Published: 1 February 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Analyzing and annotating sports footage manually can be tedious and time-consuming, so automating this process has great potential in reducing human error, saving time, and decreasing costs. This automated analysis has the potential to assist both coaches and players in understanding how their team plays and how other teams play, which can improve overall team performance. By using this automatic analysis, interested fans can also benefit by being able to understand in greater detail the many aspects of the game, enriching their game-watching experiences.

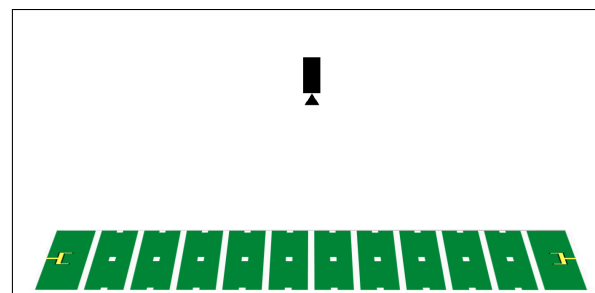
With American football (hereafter referred to as football) being an extremely popular sport, particularly in the United States, many people are interested in the analysis and statistics of the game. To analyze football footage effectively, there are several key aspects of the game that need to be identified before more complex analyses can be done. Player location, player position (role) labels, and team formation are core components that must be identified before analyzing the footage of the opponents' past games to help coaches develop game plans.

Football in particular can benefit greatly from automatic annotation and analysis. The inherent play style and physical environment present a situation that can be addressed with computer vision and deep learning. Each football team is allowed to have eleven players on the field at a time for each play. These players consistently position themselves in a somewhat predictable pattern to attempt to outmaneuver their opponent. Because of these expected aspects of the game, artificial intelligence can help predict certain patterns, which assist in the analysis of player locations and movements. Football coaches can use this knowledge to develop a game plan to improve their understanding of the game and, more specifically, how their players react to the opposing team's different formations.

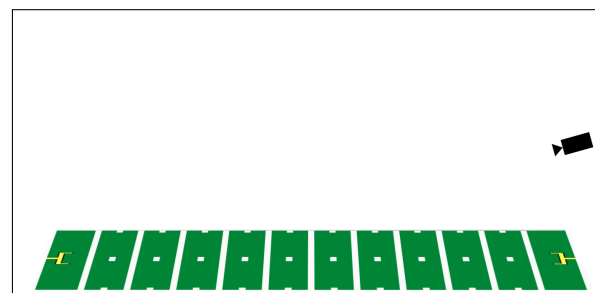
Given the nature of football footage, various challenges exist in locating players and analyzing formations. One challenge arises because of the common placement of the camera. In the camera view we are using, the play is seen from above and behind the line of scrimmage. Because of this, some players are consistently occluded. In particular, the quarterback often stands directly behind the center, prohibiting the center from being seen in the image. The defensive players closest to the line of scrimmage are also often occluded by the offensive line. Our approach to solving this issue is addressed in Section 4.3.4.

Another challenge that arises comes from the inconsistent camera placements for different teams. In some camera placements, only a portion of the players are in the frame, and many camera placements have differing angles and fields of view. This inconsistency in the footage can make it difficult to obtain homogenized data.

These issues are avoidable if all teams placed the cameras high and at the same locations to provide close to a bird's-eye view of the field, though resolving these particular issues is beyond the scope of this work. As this is a feasibility study, we have collected our own clean, consistent dataset from the Madden NFL 2020 PC game, described in Section 4.1. Because this dataset is consistent, and because the majority of players can be seen in a single frame, we do not address the issue of data from differing and inconsistent camera views in this work. A diagram of these approaches to camera placement is seen in Figure 1.



(a) Ideal Camera Placement



(b) Common Camera Placement

Figure 1. The camera placement (a) shows that the ideal location of the camera is in a position where all of the players are directly visible and no occlusion occurs; (b) shows a common placement of the camera that captures football footage is at the edge of the field, causing player occlusion.

Computer vision, artificial intelligence, and physical tracking devices are commonly used in extracting core analytical information from sports footage. The research we present takes place at the intersection of football, computer vision, and deep learning. Here, we discuss these individual fields, how computer vision and deep learning can benefit football analysis, and specific deep learning frameworks that we use in our research.

1.1. Related Work

Computer vision is one field in particular that plays a significant role in analyzing sports footage, particularly when it comes to tracking moving players. The recent work of [1–3] demonstrates the use of computer vision, combined with other techniques, in

tracking moving players in various sports. When used alongside deep learning, computer vision can produce viable results when analyzing sports footage.

Computer vision techniques were used to create a cross-domain transformation from a camera view to a bird's eye view to track the movement of American football players [3]. A Markov decision process was used to predict the trajectories of wide receivers in a football play [4]. They incorporated both prior knowledge about the game and short-term predictions of where opposing players will move. Multiple cameras were used to track soccer players [2] and track basketball players [5]. A broadcast view of the court and the play-by-play text of the game were used [6]. These works are good examples of how computer vision is being used to analyze sports footage. The above five works aimed to track players via sports footage, while the aim of our work was to extract information from a single image that could be expanded into video.

Atmosukarto et al. analyzed a video and extracted the following three elements: (1) the frame in which the offensive team is lined up in a formation (with 95% accuracy), (2) the line of scrimmage (with 98% accuracy), and (3) the type of formation of the offensive team (with up to 67% accuracy) [6]. The third element they extract, the type of formation, is most similar to the work we present here. However, there are some key differences. They used a dataset containing real footage of American football, as opposed to our current focus which uses simulated images from the Madden NFL 2020 PC game. This makes their system closer to the ideal use case of working with real footage. However, the process used in their work identifies eight total formation classes, while our process identifies twenty-five total formation classes. Additionally, their method is based on using Support-vector machine (SVM) classifiers and the histogram of oriented-gradient (HOG) feature descriptors, while ours is based on deep learning techniques. We currently achieve an accuracy of 84.8% in identifying one of twenty-five formation classes using artificial data, while they achieved up to 67% accuracy in identifying one of eight formation classes using real football footage. Our study shows that our approach has great potential to perform well on real football footage. The dataset used by [6] is publicly available. We did not run our method on this dataset due to the difference in formation classifications and different input data requirements. This comparison will be left to be done in future works.

Artificial intelligence techniques, such as deep learning and machine learning, allow for more robust player tracking in sports. Using the RFID tracking technology, reference [7] quantifies the quarterback's decision-making process, predicting which player will receive the pass. Deep learning has been used to detect a soccer ball and a Kalman filter was used to track the ball [8]. Another work also used deep learning techniques to track soccer players [9]. These works show the rising prevalence of artificial intelligence in the field of sports analytics.

1.2. Scope of Work

Automating the process of analyzing and extracting information from football footage is the premise of this research, and the combination of computer vision and deep learning provides a way for this automation to happen. We focus on detecting player locations, labeling the players with their football player label (quarterback, safety, etc.), and identifying the offensive formation. Computer vision and deep learning have the potential to assist greatly in automating sports footage annotation and analysis. These advancing technologies are particularly useful for locating and tracking objects in a visually limited environment. Many sports inherently have environmental constraints, which makes deep learning an ideal solution for sports analysis.

Sports analytics using computer vision is becoming increasingly viable, and our goal is to apply deep learning methodologies to improve the current state of the field, specifically in relation to football. One way this research aims to do this is by eliminating the need for physical tracking devices attached to the players, enabling the program to rely solely on footage obtained from a single camera. We present a method of automatically locating and labeling players, as well as identifying the offensive formation, from an overhead image of a football play before the play begins.

This work aims to provide a fundamental analysis—a prerequisite to further research. Using data obtained from the Madden 2020 PC game, we present a system that extracts data about football players and formations using both computer vision and deep learning. Player locations and labels, as well as offensive formations, are identified using an overhead camera. This work shows the feasibility of using artificial intelligence in general, and deep learning in particular, to analyze and extract useful data that can be used for football analytics. It also provides a foundation for future work in football analysis.

2. American Football

An understanding of the basic rules of football is beneficial in understanding how our football analysis system works. The game takes place on a rectangular field with end zones on either end. Two opposing teams, each with eleven players, are on the field at the same time, with one playing the offense and the other playing the defense. The offensive team attempts to move the football into the defensive team's end zone to score. This is done by arranging the players in strategic formations and either running with or passing the ball down the field past the opposing team.

All twenty-two players on the field are assigned to one of twelve positions that come with specific responsibilities. The left column in Table 1 shows these twelve positions. The first seven positions are for the offensive team and the last five are for the defensive team.

We group these twelve positions into eight-player groups or labels because the twelve original player positions do not all provide useful information for identifying offensive formation. This was done by consolidating seven of the player labels (center, offensive guard, offensive tackle, cornerback, safety, defensive end, and defensive tackle) into three player labels (offensive line, the defensive back, and the defensive line). This simplifies the work of both player labeling and formation identification.

Table 1. Twelve player positions are grouped into eight groups or labels for classification.

Player Positions	Player Labels
Center Offensive Guard Offensive Tackle	Offensive Line
Quarterback	Quarterback
Running Back	Running Back
Tight End	Tight End
Wide Receiver	Wide Receiver
Cornerback Safety	Defensive Back
Defensive End Defensive Tackle	Defensive Line
Linebacker	Linebacker

The number of running backs, tight ends, and wide receivers determines the personnel identification of an offensive formation. Table 2 shows a list of eleven common personal identifications. The number of running backs determines the first number of the personnel identification, the number of tight ends determines the second number of the personnel identification, and the number of wide receivers is understood to make up the difference of five total players, so wide receiver is not included in the personnel identification. Every formation includes a quarterback, five offensive linemen, and five players from one of the various personnel identifications to form an eleven-player team on the field.

Table 2. The personnel identification is determined by the number of running backs (RB), tight ends (TE), and wide receivers (WR) (this table is not exhaustive).

Personnel	RBs	TEs	WRs
00	0	0	5
10	1	0	4
11	1	1	3
12	1	2	2
13	1	3	1
14	1	4	0
20	2	0	3
21	2	1	2
22	2	2	1
23	2	3	0
32	3	2	0

The running back's specific location also has an important role in the offensive formation. Different terms describe the formation based on the location of the running back, such as strong (the side of the field that the running back is on), weak (the side of the field that the running back is not on), split (when there are two running backs on either side of the quarterback), and empty (when there are no running backs).

The personnel identification and the running back alignment are methods of describing specific elements of an offensive formation. By knowing these specific elements, coaches and players are able to clearly communicate and make specific changes to the formation when needed. This is where an audible in football is used, i.e., when the quarterback sees that the offensive formation is not suited against the opposing defensive formation and, thus, decides to alter the player formation.

Depending on the current state of the game, the offensive team chooses from several offensive formations. Formations make up formation families, which are groups of formations that have similar player placements. A formation defines specific placement for each of the players, while a formation family defines a more general placement of players. Table 3 shows the five formation families selected for our study. Each formation family contains five formations. We collected images for these twenty-five formations for classification from the Madden NFL 2020 PC game for our research.

Knowing the strategies and tendencies of one's own team and the opposing team can be very beneficial to a football team. With this knowledge, coaches and players can prepare for future games by practicing specific strategies, giving them an advantage over the opposing team.

Table 3. The five formation families selected for our study. Each consists of five formations.

Formation Family	Formations
I Form	I Form Close Slot I Form H Pro I Form H Slot Open I Form H Tight I Form H Wing
Pistol	Pistol Bunch TE Pistol Full Panther Pistol Spread Pistol Strong Slot Open Pistol Wing Flex
Shotgun	Shotgun Ace Shotgun Doubles Shotgun Eagle Trey Shotgun Slot Offset Shotgun Wing Tight
Singleback	Singleback Ace Double Wing Singleback Deuce Singleback Doubles North Singleback Trio Singleback Wing Pair
Strong	Strong H Pro Strong H Slot Strong H Wing Strong Tight Strong Twins Over

3. Methods

3.1. System Modules

Our system consists of three modules, each with a specific purpose: the player localization module, player-labeling module, and formation identification module. Figure 2 shows the overview of the system architecture. Each module uses a deep neural network to complete its specific task.

The player localization module processes an image of a football play immediately before the ball is snapped. The locations of the visible players for a single image are detected and passed into the player-labeling module. Each of these players is classified and assigned a player label, such as quarterback or safety listed in the right column of Table 1.

By the nature of labeling the individual players, this labeling process has the added benefit of separating the offensive players from the defensive players. The offensive player locations and labels for a single image are then passed into the formation identification module, where the offensive formation is identified as one of twenty-five formations.

The network architectures used in this work are not new. However, we develop a unique way to process the output data of the player localization module, and present it in a specific arrangement to the player labeling network. We then process the output of the player-labeling module and present it, also in a unique representation, to the formation identification module to identify the offensive formation for our unique application.

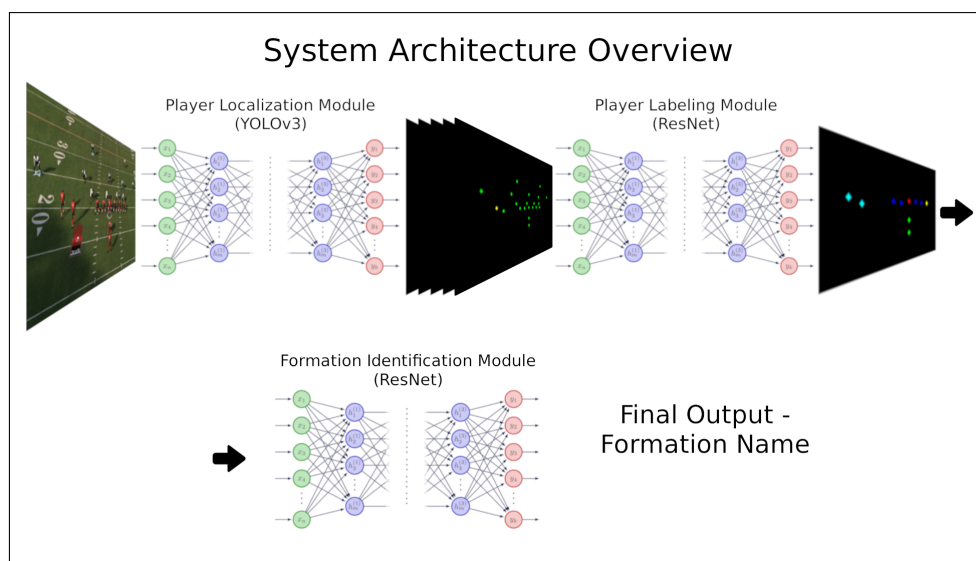


Figure 2. System architecture overview. the system consists of three modules: (1) player localization, (2) player labeling, and (3) formation identification.

3.1.1. YOLO: Player Localization Module

Our first module is the player localization module which detects the location of the visible players in the overhead image of a football formation. In order to detect player locations, we use the You Only Look Once (YOLO) deep learning framework [10]. Specifically, we use YOLOv3 [11] to locate and obtain a bounding box for each player in the image. Newer versions of YOLO have become available since we carried out this task in 2020. This framework allows for extremely fast object detection. Though we only use still images in our current research, using YOLO gives us the ability to easily extend our research to video, a potential step for future work.

The YOLOv3 architecture [11] introduced a new way of detecting objects in images. When analyzing an image, the convolutional neural network runs only a single time, which is one reason the YOLOv3 architecture is significantly faster than previous architectures. The architecture is made up of several neural network layers, mainly convolutional layers, max-pooling layers, and fully connected layers. It also contains skip connections, connections that connect later layers of the network to earlier layers to increase robustness. The image is split into multiple regions, and each region has the job of determining if an object of interest exists at that location. YOLOv3 has the added benefit of processing the image at different scales, making it a more robust object detector. If an object does exist at a given location, the output consists of the bounding box information, a confidence score, and a class.

Using the GitHub repository TrainYourOwnYOLO by Anton Meuhlemann [12], we trained a YOLOv3 architecture with our custom dataset of the images of the individual players (obtained with the bounding box coordinates collected when gathering data). It was trained for a single class, giving a player label to all of the visible players. We used YOLO because of its ability to detect players and provide a bounding box for the detected players. We include a diagram to visually identify how the YOLOv3 architecture fits into the player localization module in Figure 3.

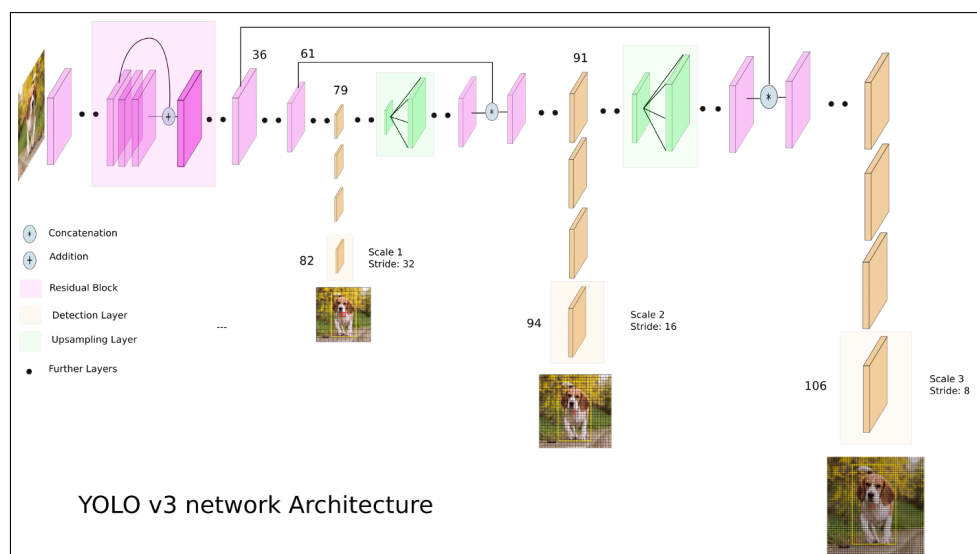


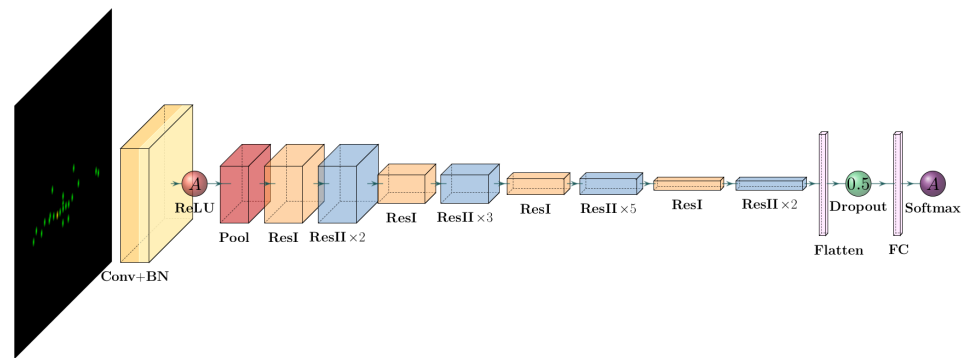
Figure 3. YOLOv3 Architecture. YOLO processes an overhead image of a football game and finds the locations of the visible players (image credit to Ayoosh Kathuria [13]).

3.1.2. ResNet: Player-Labeling Module

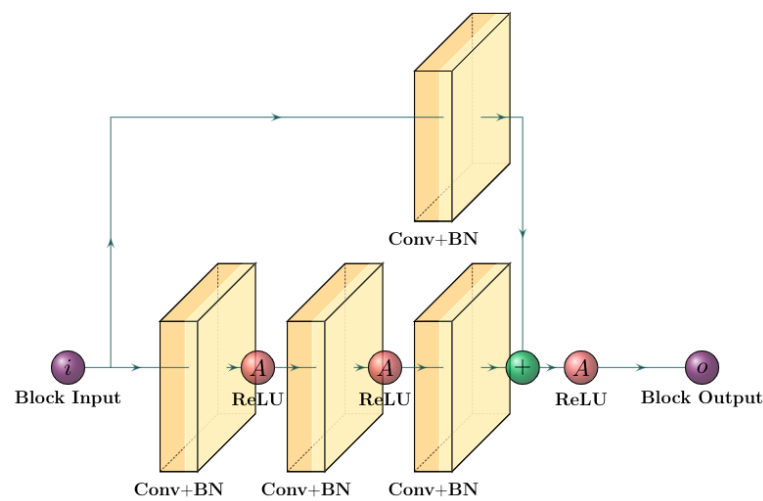
To label the individual players and to identify the offensive formation, we use a residual network (ResNet) framework [14]. We chose to use a ResNet as it is a common architecture used in computer vision because it allows for deeper neural network architectures, leading to better-performing models. In addition, the architecture excels at networks that have many layers. Before ResNets, a common problem that existed with deep neural networks was the vanishing/exploding gradient problem, as discussed in the research proposing ResNets [14]. This issue causes values in a neural network to go toward zero, leading to an unusable network. The core idea of ResNets is to use skip connections, connections that bridge later layers to earlier layers. This allows the network to become more robust by learning to detect patterns more reliably.

Because of the ability of a ResNet to create more robust deeper neural networks than was previously possible, we chose to utilize a ResNet as it would undoubtedly be capable of detecting the patterns in sports footage. In the case of labeling the individual players, a ResNet has the ability to differentiate among the different player labels based on the location of the player in relation to the other players. In the case of formation identification, a ResNet has the ability to identify the formation based on both the locations of the players, as well as their individual labels.

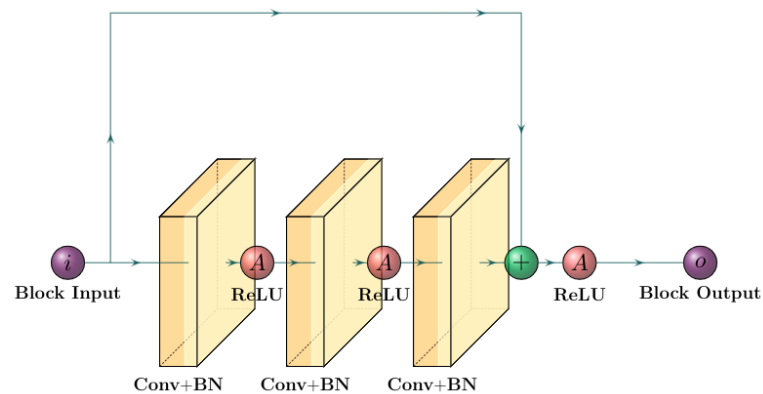
The player-labeling module classifies each player as one of eight labels, as shown in Table 1. The input to this module is the player locations output from the player localization module that includes a bounding rectangle for each player detected. The output of this module is one of those eight labels for each player. Figure 4 shows an example of the ResNet architecture and how it fits into the player-labeling module. The player-labeling module described here uses a (ResNet) that is 152 layers deep.



(a) Base ResNet Architecture



(b) ResBlock I



(c) ResBlock 2

Figure 4. Example ResNet architecture for formation identification—once the player locations are known, we use a ResNet to determine the player labels; (a) is the base architecture, (b) is Res I, and (c) is Res II.

3.1.3. ResNet: Formation Identification Module

For the formation identification module, we chose to utilize a ResNet again for the same reasons as those for using it for the player-labeling module. The ResNet for this module is essentially the same as the one for the player-labeling module and also with 152 layers deep. Unlike the player labeling model (where the input only includes player locations), this module requires the input to include both player locations and labels. The

unique representations of these two different inputs that we developed for this specific application are discussed in the next section.

The formation identification module identifies the formation of the offensive team given the location from the player localization module and the label of each player from the player-labeling module. We trained this module on five formation families of five formations each, for a total of twenty-five formations, as shown in Table 3.

3.2. Input Representations

One of our main contributions is our combination of multiple networks; unique input representation for each network comes with this combination. This is seen with the data used to train and test this system as it consists of images taken immediately before the play starts. The view is the common All-22 view used by coaches to evaluate plays, which is above and behind the offensive team. We note that the All-22 view only shows from behind the offensive team and not from the defensive side. We recognize that multiple views would increase the accuracy; however, we are confident that the bird's-eye view mentioned in Figure 1 is still the ideal case. However, as our work is to demonstrate a proof of concept, we use the All-22 view as it is currently used by coaches to evaluate players.

Our custom-made dataset was collected using the Madden NFL 2020 PC game. There were three main reasons we chose to collect the data this way. First, it gave us the ability to quickly gather data without the need to search through hours of footage on the Internet. Second, it gave us the freedom to specifically choose the plays that we needed for our dataset with their corresponding ground truth formation labels. Third, it provided fairly clean and consistent data across plays, making it easier to test the validity of the system.

The ideal camera location for this system is either directly above the play or high enough above the play to allow for all players to be seen by the camera. Figure 1a shows the ideal camera placement. Unfortunately, the common views provided by football teams have either a view from the side or from behind and above the offensive team, both of which occlude players from the camera's view. Figure 1b shows the camera placement used for this research. This does introduce the challenge of correctly extracting valid information from the footage, though our work is able to overcome this issue. The details of our solution for this challenge are discussed in Sections 4.2 and 4.3.

3.2.1. Input to the Player Localization Module

After the images were collected, we labeled each one with additional information to generate the ground truth for training and testing. Using the Microsoft Visual Object Tagging Tool (VOTT) [15], we collected bounding boxes of all visible players and single coordinates for all occluded players. Figure 5 gives an example of the data collection process and the input image to the player localization module. The bounding boxes allowed us to extract images of the individual players, which were used to train the network. The individual players were manually labeled with their corresponding player label as ground truth.

There are twelve player labels in total, with seven on offense and five on defense. The offensive labels consist of quarterback, running back, center, offensive guard, offensive tackle, tight end, and wide receiver. The defensive labels consist of defensive tackle, defensive end, linebacker, cornerback, and safety. The yard line at the line of scrimmage (the line separating the offense from the defense at the start of the play) was collected as well. This yard line information was used to augment the data for the formation identification module, as explained in Section 4.2.1).

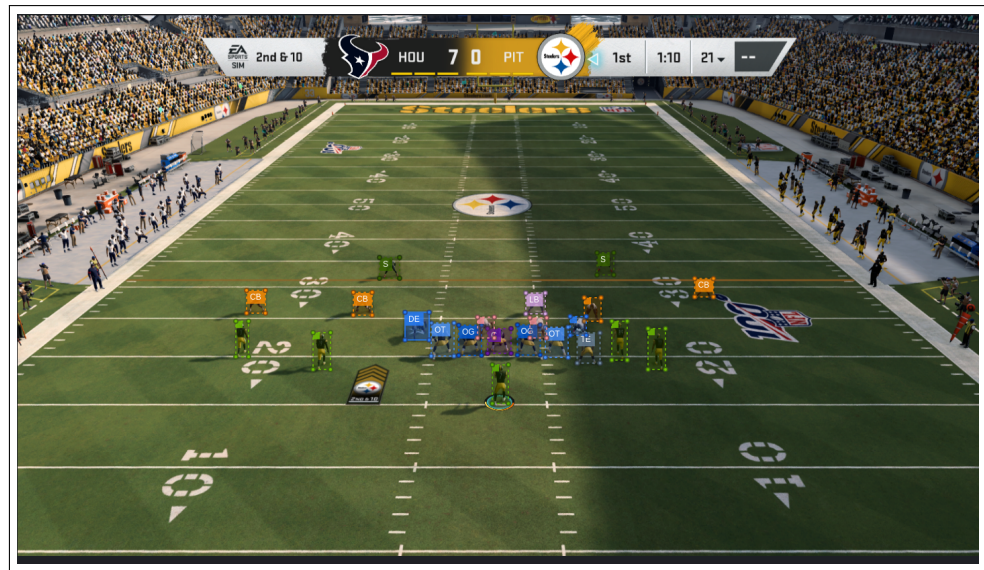


Figure 5. Annotated Data. Using the Microsoft Visual Object Tracking Tool (VOTT), we manually labeled the individual players.

Our first attempt at creating the player localization module was to train a multi-class YOLO model. This would distinguish the player labels (quarterback, safety, etc.) from one another. However, the model could not reliably differentiate between the different player labels because all players are in the same uniform, their postures are similar, and the numbers on their uniforms are often blocked or too blurry to be recognized. This led us to determine that YOLO would, in this case, be best used as a single-class model with the sole purpose of finding the location of the individual players. We moved the players' position labeling from this YOLO-based module into the next module, the player-labeling module.

The output of the player localization Module consists of a confidence score and a bounding box for each of the detected players. Because it is a single-class detector, every detected player is assigned a single-player label. A pixel coordinate is extracted from the center of each of the bounding boxes. The player locations are then passed to the player-labeling module.

3.2.2. Input to the Player-Labeling Module

Because all players are in the same uniform, their postures are similar, and the numbers on their uniform are often blocked or too blurry to be recognized, the only information that is useful for determining their labels or player positions is their locations in the image. The biggest challenge of using the ResNet to label player positions efficiently is finding a proper way to present the data to the network. As shown in Figure 6, we generate multiple input representations containing only the location information of the players for a single image. We generate one representation for each player of interest in the image. All players are assigned a green dot at their locations except the player of interest that is given a yellow dot. This unique representation of the input data allows the ResNet to classify every player in the image one at a time.

In order to minimize the amount of data going through the player-labeling module, we scale the resolution of the image down from 1920×1080 pixels to 480×270 pixels. We found that this reduction in resolution did not significantly decrease the performance of the player-labeling module, while it did significantly decrease the amount of time required to train the neural network.



Figure 6. Player labeling data—cycling through all of the players, we identify a single player at a time as the desired player to label. This player is colored yellow while all other players are colored green. A slight gradient is applied to the circular marks of all of the players to better handle players that are very close to each other, causing the marks to overlap.

For the majority of the time, the players in a formation are not centrally positioned in the image. To better process the data, we normalize the player positions within a formation. This is done by calculating the average coordinate of all of the players, determining the closest player to that average coordinate, and relocating that average player to the center of the image, as well as relocating all of the other player positions in relation to the average player. This normalizes the players in such a way that allows for more consistent processing and labeling of the players.

As training input, the network takes this processed and normalized data, as described above, as well as augmented data described in Section 4.2. As testing input, the network takes the processed and normalized data without augmentation. The network classifies the incoming data as one of the eight-player classes shown in Table 1: offensive line, quarterback, running back, tight end, wide receiver, defensive back, defensive line, and linebacker. By the nature of classifying the players using these labels, this network has the added benefit of differentiating offensive players from defensive players.

Here, we note that while the player-labeling module labeled all offensive and defensive players, for reasons discussed in the following subsection, only the offensive formation was identified by the formation identification module. This results in the defensive players that are labeled by this module not being passed onto the formation identification module.

3.2.3. Input to the Formation Identification Module

It would be possible to identify the formation of the defensive team, given the same data that exists in our collected dataset. We attempted to collect the defensive formation names to analyze alongside the offensive formations using the Madden NFL 2020 PC game. However, we found that we did not have the same level of control over choosing the defensive plays in conjunction with choosing the offensive plays. The defensive formations that were produced automatically in response to our offensive formations selection were random and cannot be specifically controlled. Because of this, the Madden NFL 2020 PC game was unusable for identifying the defense formation for our system. As college football teams are most interested in analyzing offensive formations, we focused the formation identification module solely on identifying offensive formations. This problem can be avoided with ideal camera placement as seen in Figure 1.

For the training input, the formation identification network takes data that are processed, normalized, and augmented in the same way as the player labeling network. As expected, the test input for the network is the processed and normalized data without any augmentation. The network classifies the incoming data as one of the twenty-five formations defined in Table 3.

Similar to the player-labeling module, we developed a unique way to present the data to the network. Besides the player locations that are represented by their coordinates in the image, we also color-code the players according to their positions in order to provide player labels to the network. Using the ground truth data containing location and player labeling data, we generated training and testing data for the formation identification module. Figure 7 shows an example of the generated data. Offensive linemen are dark blue, quarterbacks are red, running backs are green, tight ends are yellow, and wide receivers are light blue.

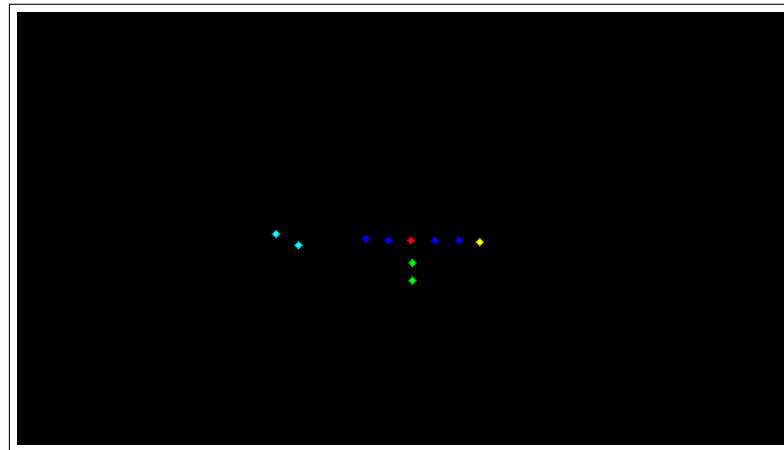


Figure 7. processed data for formation identification module—the locations of the players are represented with the pixel coordinate locations of the marks, and the labels of the players are represented by the colors of the marks.

4. Experiments

4.1. Dataset

As discussed previously, the data was collected using the All-22 in the Madden NFL 2020 PC game. For labeling the data, the names of the offensive formations were also collected for use in the formation identification module, as explained in Section 3.2.3. For a more robust understanding of how we represented the data in our dataset, please see Section 3.2.

We collected and labeled 1000 images in total from the Madden NFL 2020 PC game, each with a resolution of 1920×1080 pixels. The first 500 images were collected without the formation of ground truth. They can only be used for the player Location and player labeling networks. The second set of 500 images was collected with the formation information and can be used for all three tasks. The player locations and player positions were labeled manually to provide the ground truth.

Out of the 1000 images, 700 images were used to train the player localization network, while 300 images were used to test the network. The player-labeling module also uses data from 700 images for training and 300 images for testing. Because only 500 images were collected with the formation information, the formation identification network uses 300 images for training and 200 images for testing. The images were split into training and testing by randomly selecting certain games for the testing dataset. Each game represented different teams and the time of day of each game also varied. This allowed us to have the greatest variability within our dataset and provides an accurate training and validation split to test our results.

4.2. Data Augmentation

Artificial data augmentation provides two main benefits for our system: First, it provides us with more diverse data to train the deep neural networks, and second, it provides us the flexibility to augment the data in ways that will improve the robustness of the model. We performed four total augmentations, each of which is explained in further detail below: yard line augmentation, player-shifting augmentation, formation rotation augmentation, and player count modification augmentation. We used three of these four augmentations in our final system, deciding not to augment the player count for each formation. We further explain this decision while describing the augmentation method below.

4.2.1. Yard Line Data Augmentation

Each image included in our dataset is from a specific yard line. In reality, the same formation could be located at different yard lines all over the football field. To better diversify the formation data, we implemented yard line augmentation. This is done using

a 2D affine transformation to transform player locations to new coordinates in the image as if the image were captured when the formation is at different yard lines.

The original player location data (which includes the yard line where the line of scrimmage is located) is used as a baseline for generating the new coordinates of all players in the same formation but for different yard lines. We generate a new set of player coordinates every 10 yards by applying an affine transformation to the original location data. This affine transformation consists of scaling the relative locations of all of the players. Examples of this augmentation are shown in Figure 8.

Ideally, a full 3D perspective transform that considers a camera model with specific camera parameters would be used instead of a 2D affine transformation. This would allow for a more realistic representation of the player locations. However, because the change in distance between the players is relatively small, we decided that a 2D affine transformation would suffice.

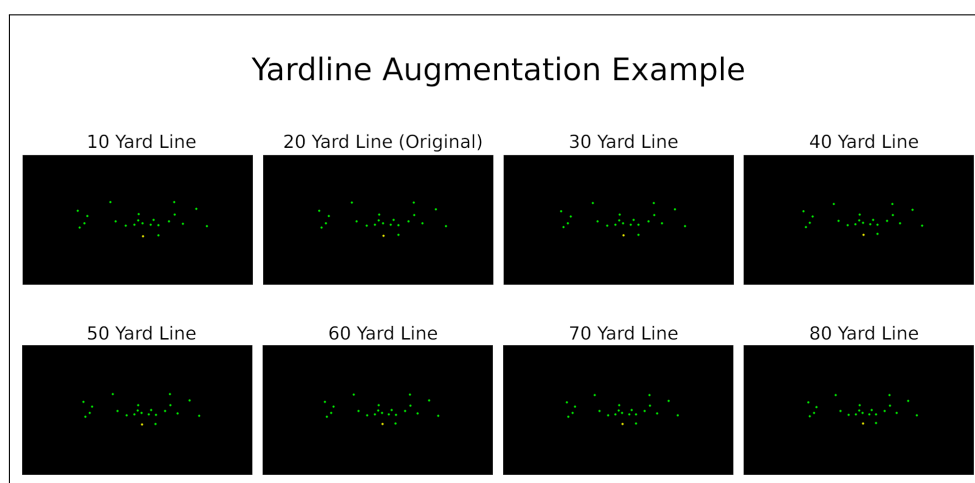


Figure 8. yard line augmentation example—the original data for these augmentations came from a formation on the 20-yard line. Formations on the 10-, 30-, 40-, 50-, 60-, 70-, and 80-yard lines are artificially augmented with the 2D affine transformation.

4.2.2. Player Shifting Data Augmentation

Players do not always stand in precisely the same location every time, even for the exact same formation. To account for this variation in player position, we implemented player-shifting augmentation. This consists of shifting the players in random directions within a range of a random number of pixels. Examples of this augmentation are shown in Figure 9.

4.2.3. Formation Rotation Data Augmentation

Formations are not always completely horizontal in the image, so in order to account for this, we implemented our formation rotation augmentation. We display two examples of this augmentation in Figure 10. This augmentation consists of rotating all players around a single average player by two degrees in both directions.

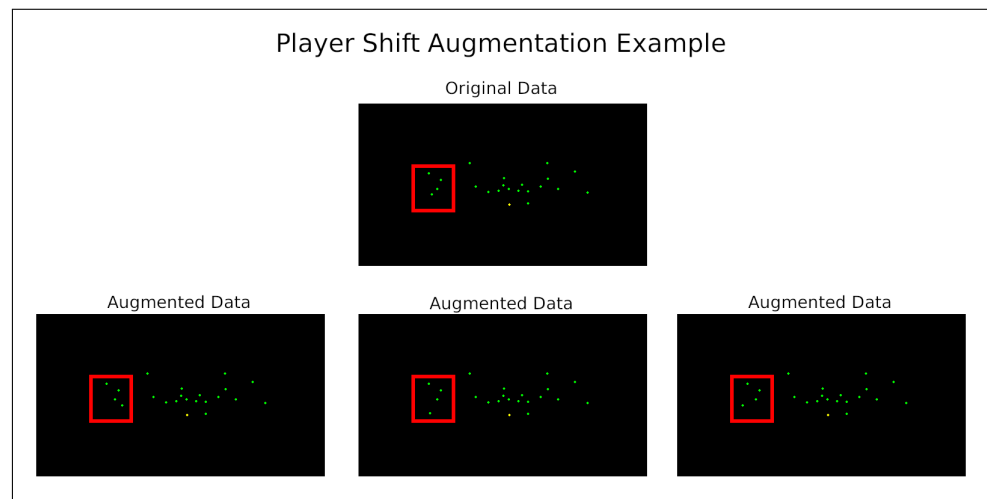


Figure 9. player shifting example—the original player can be shifted multiple times within a specified range of pixels.

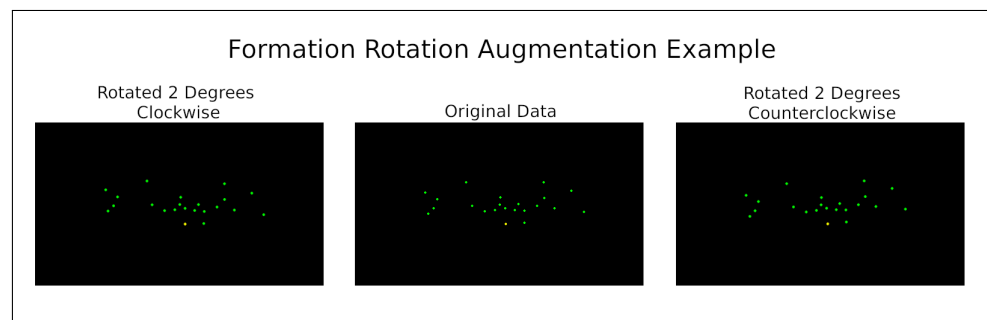


Figure 10. Formation Rotation Example—using the original data as a baseline, we rotate the players in two-degree formations (both positive and negative).

4.2.4. Player Count Modification Data Augmentation

The player localization module occasionally misses some players, so this augmentation attempts to account for those missing players when training the formation identification network. We do this by randomly adding and removing players from the original data. The data augmentation can be seen in Figure 11. We did not use this augmentation in our final system as it actually decreased the system's ability to correctly identify the formation.

We believe this to be the case because the ability to correctly identify the formation depends heavily on the presence and location of key players (mainly the wide receivers, tight ends, and running backs). Some of the formations are very similar, with only a one- or two-player difference between them. Figure 12 shows two similar formations. By adding or removing even a single player to some formations, the training data fails to teach the network consistently. In football, as opposed to soccer, there is no possible way to have more or less than 11 players on the field. With this fact, along with our training results, we removed this augmentation method from our training process.

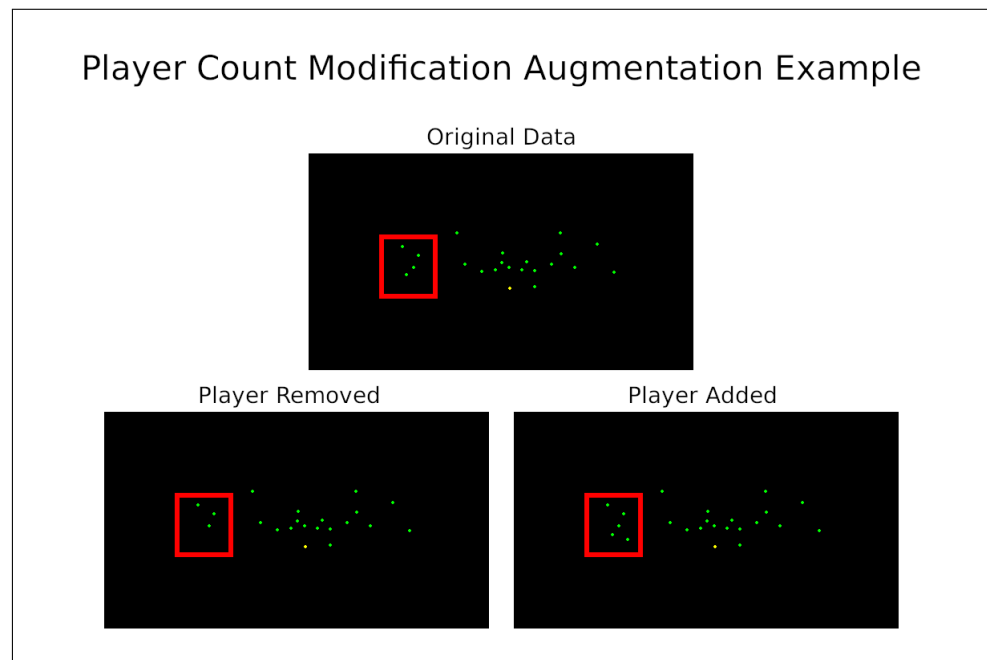
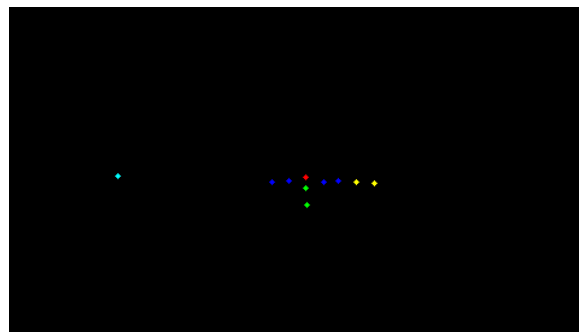
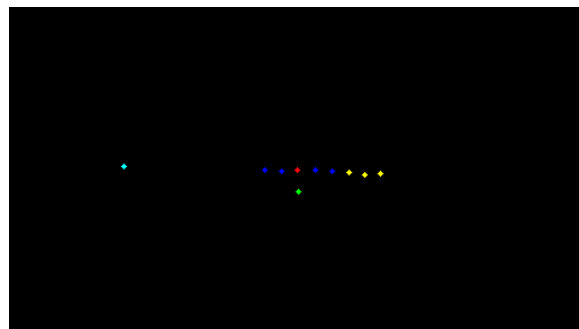


Figure 11. Player count modification example—players are randomly added or removed from the data.



(a) I Form H Wing Formation



(b) Singleback Wing Pair Formation

Figure 12. Similar formations—these two formations are very similar with only a one-player difference between the two images: one of the running backs in (a) is replaced with a Tight End in (b).

4.3. Investigations

Throughout the development of this system, we investigated multiple different approaches as potential solutions. In this section, we describe these approaches and whether or not we included them in our final system along with the reasoning behind these decisions.

4.3.1. Dataset Size

While collecting data for the player localization module, we used the precision–recall metrics with a confidence score of 0.35 to determine the preferred number of images for training. We collected the precision–recall metric results after training the model on various amounts of data (250, 300, 350, and 400 images). Using a confidence value of 35% for the threshold in the player localization module, we obtained a Precision of 97.65% and a Recall of 91.81% using 400 training images. The results are shown in Figure 13. This clearly shows the benefits of adding data up to 400 images and also displays that the rate of improvement decreases past this point. This is seen with the increase from 350 to 400 images being significantly less than the increase at prior intervals. This demonstrates the well-known principle of deep learning that an increased amount of data will result in a more accurate result. Our dataset consists of 1000 images in total, though the additional data is of most benefit to the later modules.

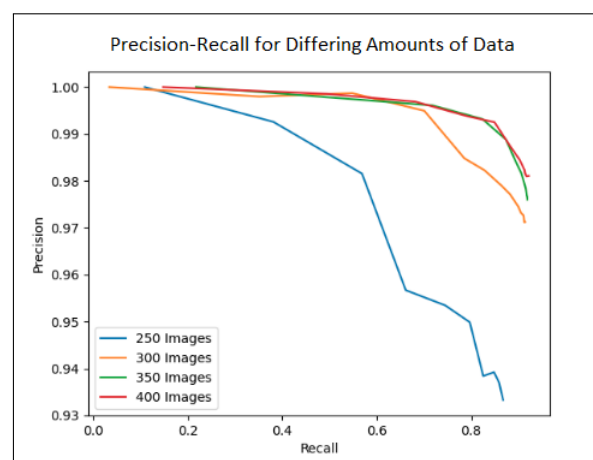


Figure 13. Precision–recall results—we trained multiple models on differing amounts of data (250 images, 300 images, 350 images, and 400 images).

4.3.2. Known Offense and Defense

We initially processed the data for the player-labeling module as shown in Figure 14. This method assumes that the offensive and defensive players are known. Twenty-two data points were created from each ground truth image collected. Cycling through each player’s location as a “root” location, we determined the root player (shown in yellow), the root player’s team (shown in green), and the root player’s opposing team (shown in light blue). This was done by creating three channels: a root channel, a team channel, and an opposing team channel. These three channels were combined into a single piece of data, which was then classified into the eight corresponding player labels.

This method assumes that the offensive and defensive players are given from the player localization module. Because the player localization module is used only to locate the individual players, we did not use this labeling method.

4.3.3. Number of ResNet Layers

We tested different numbers of layers in the ResNet module used for the player-labeling module. The different number of layers we tested were 50 layers, 101 layers, and 152 layers. We used 152 layers because, although it has the most layers and is more computationally expensive than the other options, it increased the performance of the player-labeling module. As a system for deploying this network would likely not be limited to a low computationally embedded device, but rather have the option of high-end processors or a GPU, the trade-off for higher accuracy in exchange for higher computational intensity is acceptable.

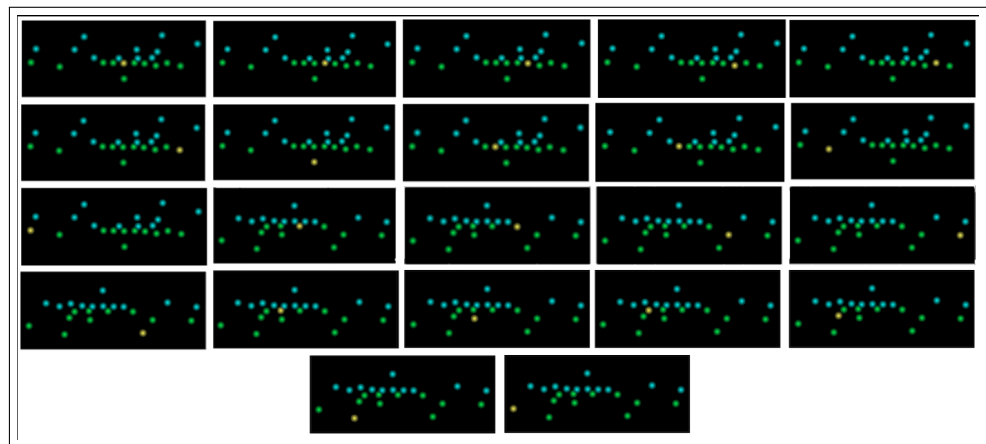


Figure 14. Original player labeling Data (zoomed in for a clearer view)—this is an example image of the data going into the player-labeling module with knowledge of the offensive and defensive players (the images with a defensive root player have been turned 90 degrees to better show the difference between the offense and defense).

4.3.4. Special Rules

Previously, we discussed the non-ideal camera view as seen in Figure 1. In this view, the camera is located above and behind the offensive team on the line of scrimmage, at an angle of about 30 degrees. This very often results in occluded players. Figure 15 shows an example of this situation. This player occlusion can be avoided if the camera height is raised to about 45 degrees or higher in the real-world setup.

The I-form formation family in particular is one set of formations that are negatively impacted by occlusion. These formations consist of four players lined up directly behind one another near the center of the formation as shown in Figure 15. Because the player localization module has a difficult time locating all four of these players due to occlusion, the overall accuracy of our system was negatively impacted. In order to account for occluded players in this system, we implemented three special rules, explained here. It is important to note that in real-world systems using a working version of this system, we will require the camera to be overhead, thus, removing the issue of occlusion. These rules are required due to the limitation of camera positions in the Madden NFL 2020 PC game that was used to collect the initial dataset.



Figure 15. example of player occlusion—the center is directly in front of the quarterback, who is directly in front of two running backs. The player localization module has a difficult time locating all four of these players because of occlusion.

Quarterback Rule

This rule ensures there is always one quarterback in the incoming formation data. If the incoming formation data does not contain a quarterback, then a quarterback is inserted at the center of the image. If the incoming formation data contain more than one quarterback, all but one are removed, keeping the quarterback that is closest to the center of the image. Adding this rule improved the formation accuracy by four percentage points. Due to this improvement, we implemented this rule into the formation of data processing identification.

Offensive Line Rule

This rule ensures there are always five offensive linemen in the incoming formation data. If the incoming formation data contains less than five offensive linemen, an offensive lineman is inserted directly in front of the quarterback (there are only ever four or more offensive linemen detected, with the missing lineman occluded by the quarterback, so the only addition needed in these cases is a single offensive lineman). If the incoming formation data contains more than five offensive linemen, the players who are most likely to be on defense (the players with a lower y-coordinate pixel value) are removed. Figure 16 illustrates the idea behind this rule well. As adding this rule did not improve the accuracy of the system, we did not implement it into our formation identification data processing.

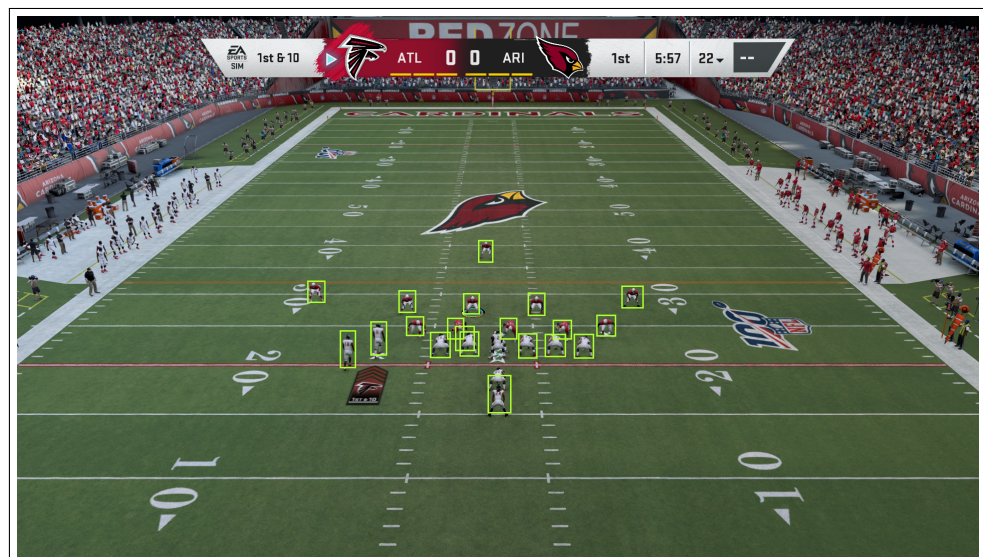


Figure 16. The output from the player localization module fails in a case that would affect the number of defensive linemen. There is an extra player identified on the left between an offensive and defensive player, thus motivating the evaluation of the offensive line rule. Occlusion is also evident in this image for one of the linemen in the center of the formation.

Running Back Rule

While testing the results of the formation identification module, we found that a majority of the misidentified formations originated from the same formation family. In one test, we found that out of the 28 formations that were misidentified, 20 of them originated from the I-form formation family. Visually analyzing these misidentified formations, we discovered that nearly all of them failed to include all four of the key players (two running backs, the quarterback, and the center).

We assumed that if the I-form formations being passed into the formation identification module included two running backs, the formation identification module would be able to correctly identify the formation. To test this assumption, we inserted a second running back directly behind the quarterback in every I-form formation and recalculated the overall accuracy at 94.0%.

In the final evaluation of the system's overall accuracy, we do not apply the rule just described because of the use of outside information (knowledge of what formations to apply the rule on). However, this test does show that the I-form formations are the cause of a significant decrease in overall accuracy (from 94% to 84.8%), and that without occlusion caused by the placement of the camera, or with a more robust player localization module, this issue would be negligible.

To overcome this, we initially implemented a rule that checks if there is a missing offensive player, and if there is, we insert a running back directly behind the quarterback. However, the number of players did not seem to be a reliable method of implementing this rule because many of the other formations also had missing players. We also attempted to implement a rule that checks if a running back exists behind the quarterback, and if one does exist, a second running back is inserted behind the quarterback. While doing this improved the accuracies of the I-form formations, it decreased the accuracies of the singleback formations (which only had single running backs behind the quarterbacks). Because of this, we did not implement this running back rule.

4.3.5. Identifying the Formation without Player Labels

In consideration of combining the second and third modules into one module, we tested the assumption that if the formation identification module were given only the player location data, it could adequately identify the formations. In other words, it would not be given knowledge of the individual player labels. This assumption is based on the idea of limiting the number of modules by combining the player-labeling module and the formation identification module in an attempt to minimize complexity. To test this assumption, we trained a formation identification model on only offensive player locations without player labels. Figure 17 shows an example of an input image without color-coded player label information. Doing this decreased the accuracy by three percentage points. We concluded that the player labels did increase the performance of the formation identification module and, therefore, decided to keep both the locations and the labels when training the formation identification module.

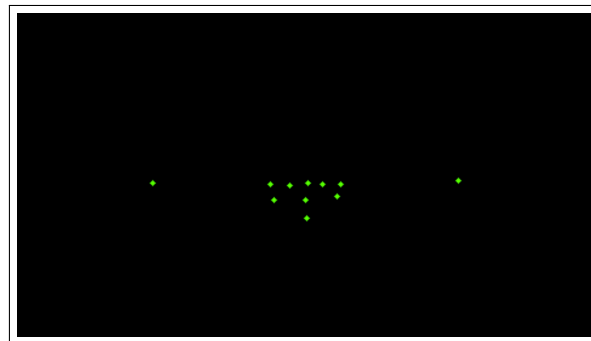


Figure 17. Formation identification data without player labels—in this experiment, the color-coded player labels have been removed from the training data, testing whether the player label information is valuable to the formation identification module.

5. Results

In order to perform a robust analysis of our system, we first evaluated the performance of each module individually using the ground truth from data collection in Sections 5.1–5.3, respectively. We discussed the challenges of evaluating the combination of the first two modules in Section 5.4; we presented our evaluation from combining the player labeling and the formation identification modules utilizing the ground truth data in Section 5.5. Finally, in Section 5.6, we evaluated the overall performance of the combination of all three modules using raw images as input and compared the results to the ground truth data.

5.1. Performance of Player Localization Module

To analyze the accuracy of a single image, we compared the ground truth visible player with the closest bounding box detected by the player's localization module. If the centers of the two bounding boxes are within 20 pixels of each other, it is considered a valid player detection. If the centers of the two bounding boxes are farther than 20 pixels, it is not considered a detected player and is excluded from further processing. If after iterating through all of the ground truth visible players, there are still player detections. These are considered incorrect detections, which represent either duplicates or false positives. Figure 18 displays an example of a graphical representation of the analysis for a single image.

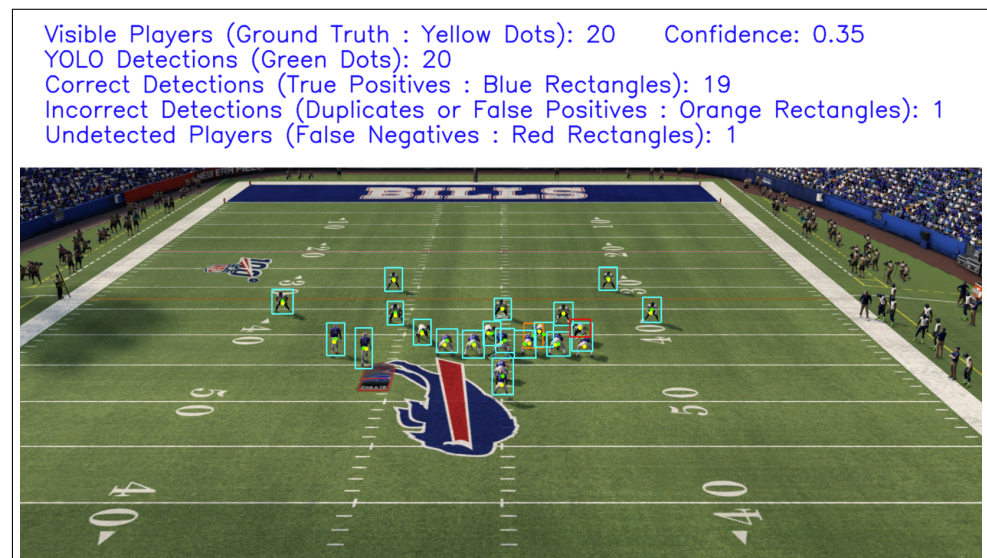


Figure 18. Analysis of player localization Module—The ground truth visible players and the players detected by the module are marked with a yellow and green dot, respectively. The correct detections are marked with a light blue rectangle, the incorrect detections are marked with an orange rectangle, and the players that are missed are marked with a red rectangle.

We calculated the total accuracy, the accuracy of the key offensive players (which consists of the quarterbacks, running backs, and wide receivers), and the accuracy of the key defensive players (which consists of the defensive backs). These accuracies are 90.3%, 94.1%, and 88.1%, respectively. These results show that the offensive players are more reliably detected than the defensive players.

We also calculated the precision–recall metrics for the player localization module. Doing this gave us an understanding of how well our model performed. In our case, precision refers to the number of correctly labeled players out of the number of detected players, while recall refers to the number of correctly detected players out of the number of visible ground truth players.

To calculate the precision–recall of the player localization module, we used the confidence score output by the module for each player. Iterating the confidence scores from 0.05 to 0.95, we calculated the precision–recall for each confidence score. If a detected player's confidence score is lower than the confidence score being tested, it would be ignored by further analysis. The remaining detected players are analyzed as shown in Figure 18. The number of detected players, the number of correctly detected players, and the number of visible ground truth players are retrieved and used to calculate precision–recall, as explained above.

We found that a confidence score of 0.35 maximized the precision and recall metrics, so we set the confidence score threshold to 0.35 for further analysis. Using this threshold,

we determined the ideal amount of data to collect, this is also explained in greater detail in Section 4.3.1.

5.2. Performance of Player-Labeling Module

We calculated the accuracy results of the player-labeling module and found that this module was able to successfully identify offensive players with a 99.9% accuracy. Out of 2075 offensive players, only 2 players were labeled as defensive linemen. Among the 2075 offensive players, 2050 players were labeled with the correct offensive player labels, giving an accuracy of 98.8%. Figure 19 shows the corresponding confusion matrix. One defensive back and one defensive lineman were mislabeled as the wide receiver and offensive lineman, respectively. One offensive lineman and one quarterback were mislabeled as the defensive linemen. The defensive players were not labeled with high accuracy. This is mostly due to player occlusion and more flexible variations of defense formations.

These results show that if the input to the player-labeling module is valid, the network can reliably label the individual players with the correct player labels and differentiate offensive players from defensive players.

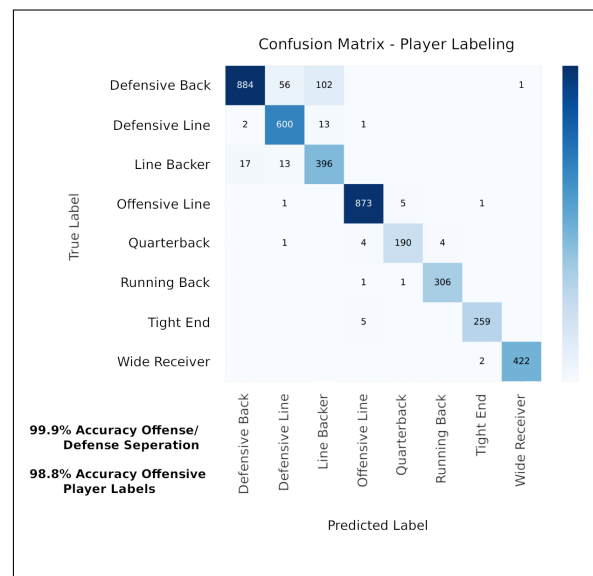


Figure 19. Confusion matrix of player-labeling module—the player-labeling module reliably differentiates offensive players from defensive players. Out of 2075 players predicted to be on offense, only 2 players were originally on defense, giving 99.9% accuracy in separating offensive players from defensive players. Additionally, out of the 2075 players predicted to be on offense, 2050 players were classified with the correct player labels, giving 98.8% accuracy in the offensive player labels.

5.3. Performance of Formation Identification Module

We utilized cross-validation for testing the formation identification module as we have only half as many images as the other modules. Cross-validation is a common method of measuring the robustness of a deep learning model and is particularly useful in cases where data is limited. Doing this gives a better understanding of how well a model will perform because it is trained on multiple sets of data. We used a value of k = 3 to split our dataset three ways and train three models. We then did a final evaluation of all three models on the same test dataset. The accuracy of each of the three models on this final evaluation is 100%, 98.5%, and 99.0%. This gives a combined accuracy of 99.2% with a standard deviation of 0.62%. Figure 20 shows the results of the three individual models. This demonstrates that as long as the input to the formation identification module is valid, the network can reliably identify the correct formation.

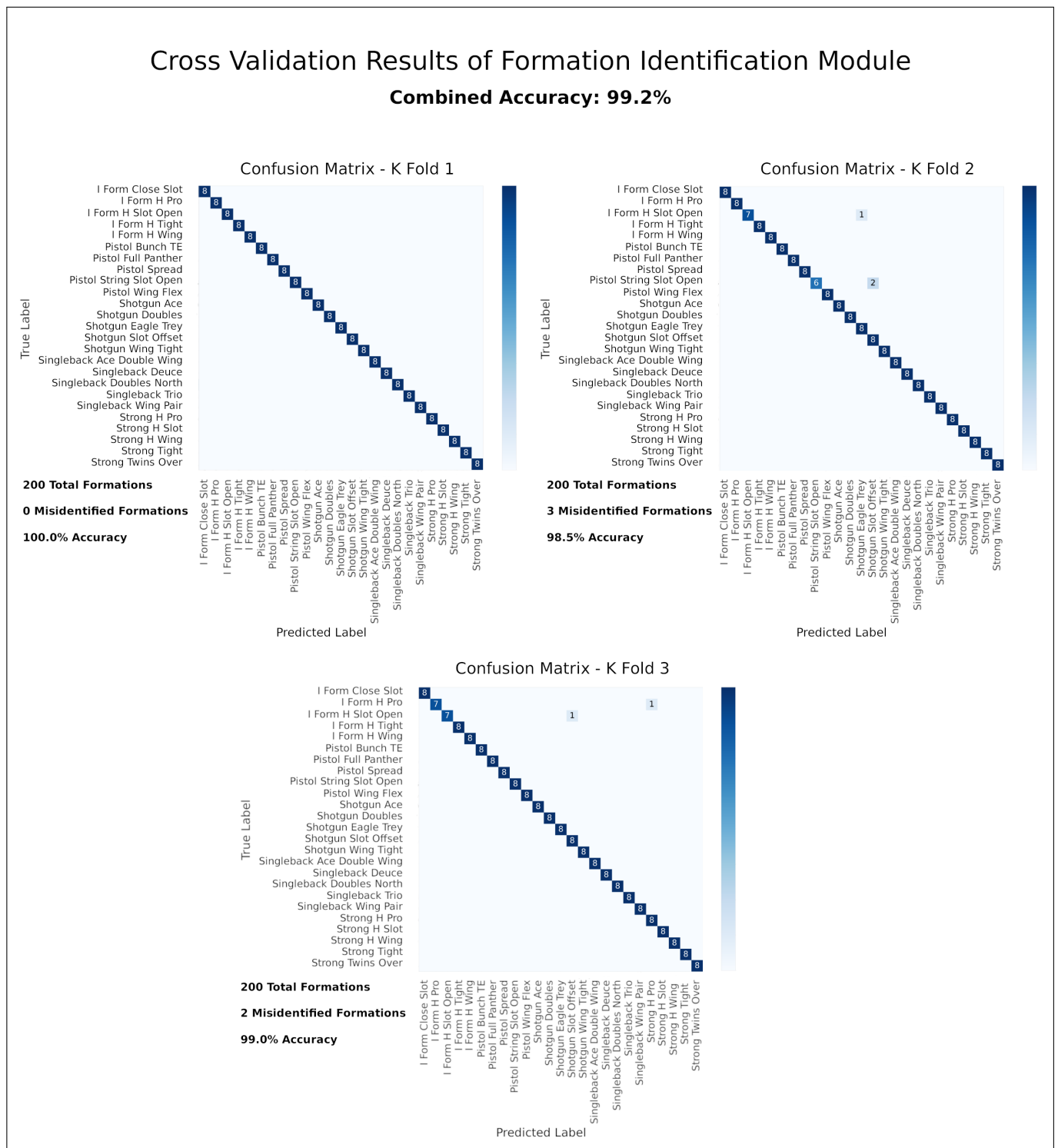


Figure 20. Cross-validation results of formation identification module—the combined accuracy of the three-fold cross-validation split is 99.2% with a standard deviation of 0.62%.

5.4. Combining Player Localization and Player-Labeling Modules

We attempted to analyze the accuracy of the player-labeling module when the input comes directly from the player localization module. However, this did not produce reliable results. After the formation is output by the player localization module, it contains new detections of players that do not have a ground truth player label. In order to reliably analyze the results of the player-labeling module in this way, we would need to manually label the players detected by the player localization module. This would take a significant

amount of time and not assist in identifying the formation. We, therefore, decided not to include the results of the player-labeling module when the input comes from the player localization module output.

5.5. Combining Player Labeling and Formation Identification Modules

The data used to evaluate the combined player labeling and formation identification modules comes from the ground truth data as explained in Section 3.2.1. Using the data as input to the player-labeling module allows us to investigate the performance of our player labeling and formation identification modules alone, isolating them from the performance of the player localization module. This ground truth data is used as input to the player-labeling module. The output of the player-labeling module is then used as the input to the formation identification module. The results of these two combined modules are shown in Figure 21. We obtain a 99.5% formation identification accuracy with the two combined modules.

We recognize that the combination of the second and third modules, when tested with ground truth player locations, actually achieved higher results than either the player-labeling module or the formation identification module. As we inspected the results, we discovered that this improvement happened because even when the player-labeling module misidentified a player, the formation identification module was still able to correctly identify the formation.

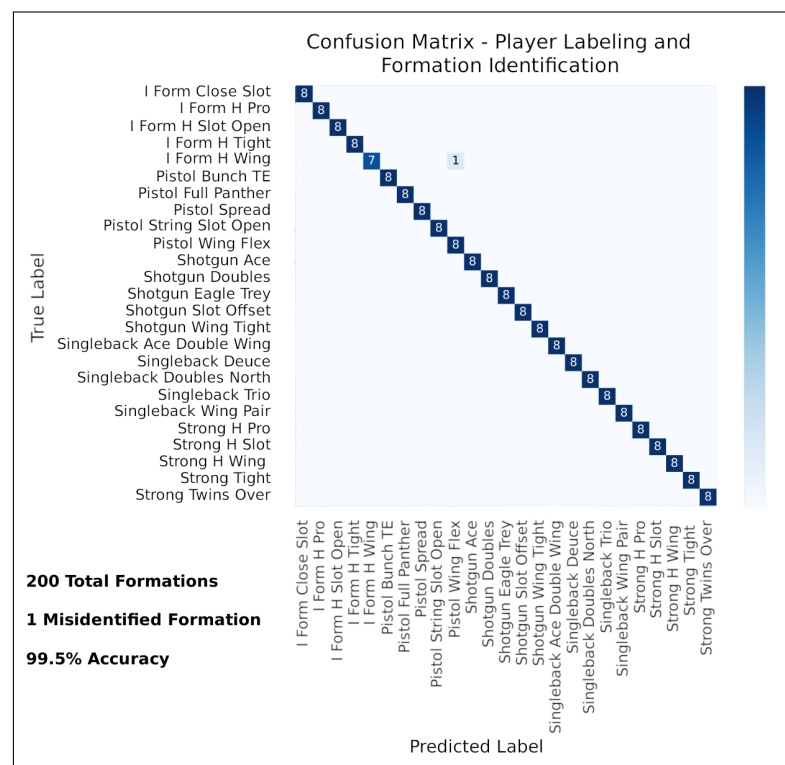


Figure 21. Confusion matrix of player labeling and formation identification modules—out of the 200 images used for testing the combined player labeling and formation identification modules, 1 image was misidentified, giving a 99.5% accuracy.

5.6. The Complete System with All Three Modules

The combination of all three modules demonstrates the overall accuracy of the system as a whole. The data used to evaluate all three modules is also the ground truth data obtained from the collected dataset described in Section 3.2.1. The data (raw image) are used as the input to the player localization module whose output is given to the player-labeling module and then that output is passed into the formation identification module.

We used cross-validation to analyze the results of all three combined networks in the same method used to evaluate the formation identification module. We did this by testing three models of the formation identification module, keeping the models for the player localization module and the player-labeling module the same. We obtained a combined accuracy of 84.8% with a standard deviation of 1.8%. Figure 22 shows these results.

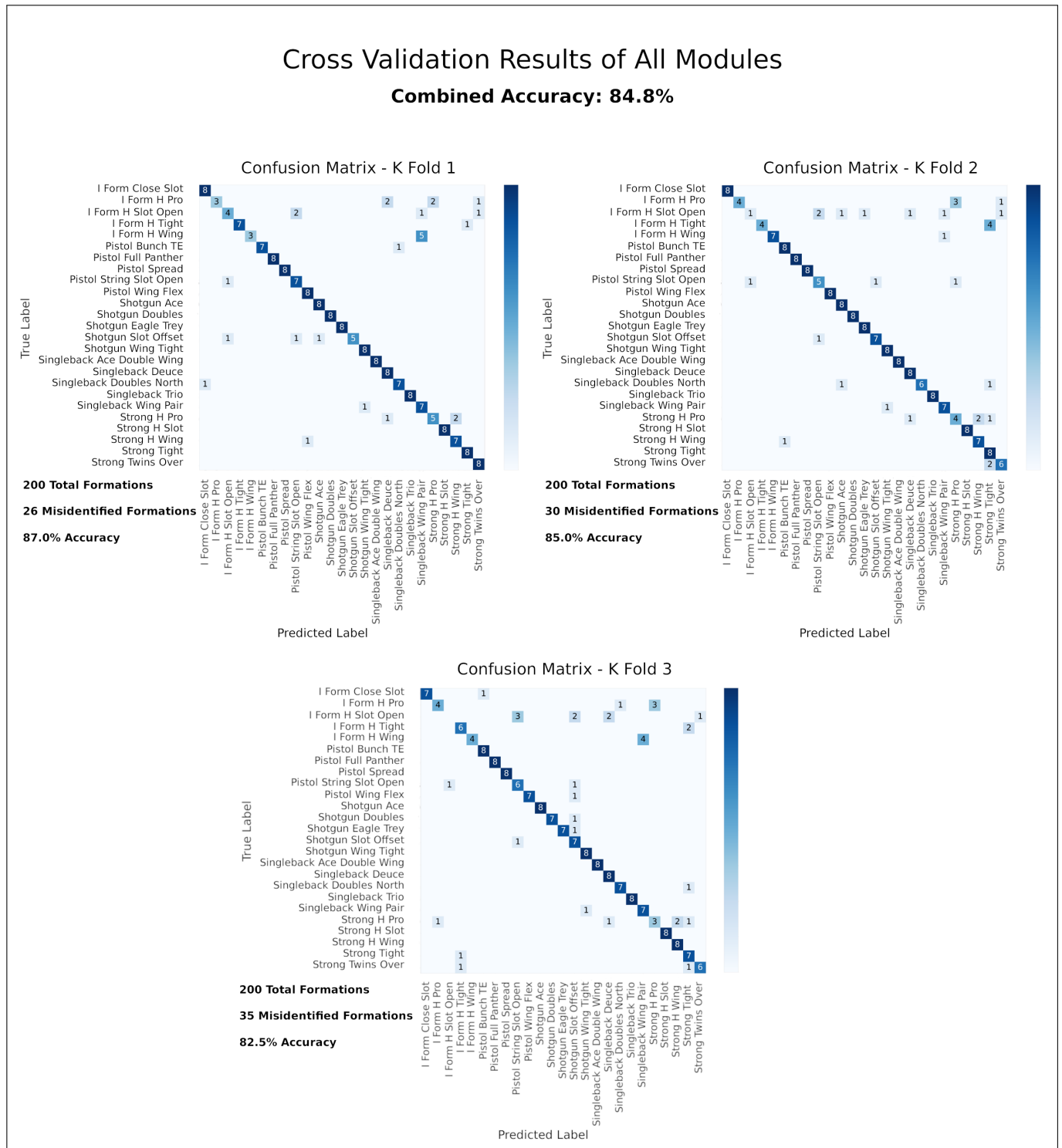


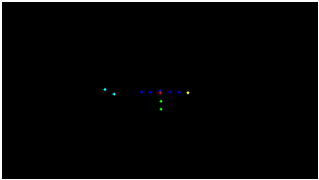
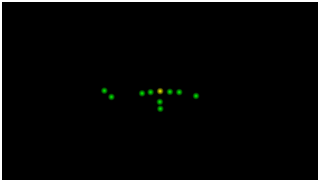



Figure 22. Cross-validation results of all modules—the combined accuracy of the three-fold cross-validation split is 84.8% with a standard deviation of 1.8%.

The overall accuracy of 84.8% is lower than the performance of the previously described modules. The drop in accuracy from 99.5% for the combination of the first two subsystems to 84.8% for all three systems is due to the inability of the player localization module to locate occluded players, which is then propagated through the other networks. The error propagating from the first network to the second and third networks makes sense in this scenario as it is impossible to correctly identify a player’s position or formation if the player is never localized. We compare the overall results of our system in Table 4. With the consideration of occluded players, this is a remarkable accomplishment. We discuss this further in Section 4.3.4.

Table 4. This table describes the overall accuracy of various methods in the system. The final row in the table represents the overall accuracy of the entire system. For an analysis of why the accuracy went up for the combination of the second and third modules please see Section 5.5.

Method	Input	Accuracy
Yolov3: Player Localization Module	 Raw Image	90.3%
ResNet: Player-Labeling Module	 Player Localized Image	98.8%
ResNet: Formation Identification Module	 Player Labeled Image	99.2%
ResNet: Player Labeling Module and ResNet: Formation Identification Module	 Player Localized Image	99.5%
Yolov3: Player Localization Module and ResNet: Player Labeling Module and ResNet: Formation Identification Module	 Raw Image	84.8%

6. Future Work

The system described here focuses mainly on detecting the locations of the players, giving each of them a label, and determining the formation of the offensive team. This alone is valuable, but knowledge about the offensive formation provides additional information.

The personnel identification and the running back alignment, as described in Section 2, are valuable pieces of information for the coach and players. This additional information is derived directly from the identified formation. The resulting information obtained from a single overhead view of a football formation consists of the locations of the visible players, labels for the visible players, the identified formation and formation family, the personnel identification, and the running back alignment.

One issue that we do not face because of our custom dataset is the presence of referees on the field. Our dataset does not contain referees on the field. Currently, the player localization module only looks for players. If a referee were on the field, our system would likely detect it as a player. However, referees wear a special black and white striped uniform that distinguishes them from the players. A simple way to ignore referees would be to implement a uniform check on all detected players. If a detection with a black and white striped uniform were found, it would be excluded from further analysis.

An additional improvement could be to streamline the player-labeling module. Currently, every individual player label requires a single run of the network. This means that an image with twenty-two visible players requires twenty-two runs of the player labeling network. It would be beneficial to implement an improved player-labeling module that takes the location of all visible players and labels them all in a single run of the network. As the current system requires running this module multiple times, once for each player, it will result in a major computational burden to run this method. We note that ResNet is likely a much larger network than the one that is required for this step. This computational burden could also be improved by replacing one or both of the ResNets with a more shallow CNN or a different machine learning approach. A further improvement to computation (and likely accuracy) that will be explored in future works is normalizing the images around the formation, such that there is little to no black space around them. This would reduce the image size, allow for a smaller network, and reduce unnecessary information fed to the network.

As mentioned, we resized all of the images for player localization to 480×270 to reduce the computation required by YOLO. Further experiments to find the ideal resolution could prove very beneficial to computation and possibly accuracy. Reducing the resolution could not only benefit the player localization module but the other two modules as well. Due to the small amount of information passed into the player labeling and formation identification modules, downsizing the resolution could have immense performance improvements.

The system described here extracts key information from a single image, but the ability to track players as they move is also a vital aspect of obtaining even more valuable information. For example, before the ball is snapped, the quarterback can call an audible and modify the formation. A single image cannot account for this change, while a video would be capable of accounting for this change. This will be overcome in future works by analyzing sequences of frames in place of single images. As mentioned previously, Atmosukarto et al. explored video analysis of real gameplay footage [6]. Further advancing our research to perform video analysis as well as analysis of real gameplay footage will enable us to evaluate our method on the same dataset and thus validate our methods.

Perhaps one of the largest improvements that could be made focuses on the overall structure of our solution. Our solution has three submodules that each perform unique, defined tasks. However, each layer in the network is arguably larger than necessarily needed for each task. To overcome this, we could combine two (or all three) of the networks into one single network. The network bypasses the player localization and player labeling and focuses on outputting the formation of the original image as the input. Such a network would undoubtedly still learn similar tasks, such as identifying players or perhaps even player positions, but this information will be identified throughout the network and provide the user with a single network to run. Potential networks that could accomplish this task include MaskRCNN, SegNet, and a transformer-based semantic segmentation framework.

One of the reasons we used a custom dataset was to overcome the challenges introduced by differing views of camera placements. A step that could be taken to introduce the analysis of real football footage would be to only use footage that meets certain requirements. These requirements would need to ensure that (1) all twenty-two players be located within the camera frame, and (2) the viewing angle meets some basic threshold that minimizes or eliminates occlusion. Implementing these basic requirements would allow the system to be used with real football footage.

7. Conclusions

The purpose of this research was to introduce a system capable of conducting automatic analysis of American football footage using computer vision and deep learning techniques. Specifically, we showed that computer vision and deep learning are valid tools for detecting player locations, labeling the individual players, and identifying the offensive formation. We showed that this is possible using a clean dataset, and with additional work, a similar system trained on real football footage could be made. We did this by creating a three-module system: player localization, player labeling, and formation identification with individual accuracies of 90.3%, 98.8%, and 99.2%, respectively. The combination of the three modules achieved 84.8% in taking a raw image and labeling the player's formation, exceeding prior published work on this subject. This work provides a good groundwork for future works that will build on this to bring about ideas and systems that improve American football analyses in the real world.

Author Contributions: Conceptualization, D.-J.L.; Methodology, D.-J.L.; Software, J.N.; Validation, J.N., A.S. and S.T.; Formal analysis, J.N., A.S. and S.T.; Investigation, J.N.; Resources, D.-J.L.; Writing—original draft, J.N.; Writing—review & editing, A.S., S.T. and D.-J.L.; Supervision, D.-J.L.; Project administration, D.-J.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lu, W.; Ting, J.; Little, J.J.; Murphy, K.P. Learning to Track and Identify Players from Broadcast Sports Videos. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 1704–1716. [[CrossRef](#)] [[PubMed](#)]
2. Koutsia, A.; Nikos, G.; Dimitropoulos, K.; Karaman, M.; Goldmann, L. Football player tracking from multiple views: Using a novel background segmentation algorithm and multiple hypothesis tracking. In Proceedings of the International Conference on Computer Vision Theory and Applications, Barcelona, Spain, 8–11 March 2007; pp. 523–526.
3. Zhang, T.; Ghanem, B.; Ahuja, N. Robust multi-object tracking via cross-domain contextual information for sports video analysis. In Proceedings of the 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Kyoto, Japan, 25–30 March 2012; pp. 985–988. [[CrossRef](#)]
4. Lee, N.; Kitani, K.M. Predicting wide receiver trajectories in American football. In Proceedings of the 2016 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Placid, NY, USA, 7–10 March 2016; pp. 1–9. [[CrossRef](#)]
5. Ming Xu.; Orwell, J.; Jones, G. Tracking football players with multiple cameras. In Proceedings of the 2004 International Conference on Image Processing (ICIP '04), Singapore, 24–27 October 2004; Volume 5, pp. 2909–2912. [[CrossRef](#)]
6. Atmosukarto, I.; Ghanem, B.; Saadalla, M.; Ahuja, N., Recognizing Team Formation in American Football. In *Computer Vision in Sports*; Moeslund, T.B., Thomas, G., Hilton, A., Eds.; Springer International Publishing: Cham, Switzerland, 2014; pp. 271–291. [[CrossRef](#)]
7. Burke, B. DeepQB: Deep learning with player tracking to quantify quarterback decision-making & performance. In Proceedings of the 13th MIT Sloan Sports Analytics Conference, Boston, MA, USA, 1–2 March 2019.
8. Lhoest, A.U. Deep Learning for Ball Tracking in Football Sequences. Master's Thesis, Université de Liège, Liège, Belgium, 2020.
9. Ma, Y.; Feng, S.; Wang, Y. Fully-Convolutional Siamese Networks for Football Player Tracking. In Proceedings of the 2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS), Singapore, 6–8 June 2018; pp. 330–334. [[CrossRef](#)]
10. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. *arXiv* **2015**. [[CrossRef](#)]
11. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**. [[CrossRef](#)]

12. Meuhlemann, A. TrainYourOwnYOLO: Building a Custom Object Detector from Scratch. 2019. Available online: <https://github.com/AntonMu/TrainYourOwnYOLO> (accessed on 30 January 2023).
13. Kathuria, A. What's New in YOLO v3? *viso.ai*, 23 April 2018. Available online: <https://towardsdatascience.com/yolo-v3-object-detection-53fb7d3bfe6b> (accessed on 30 January 2023).
14. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. *arXiv* **2015**. [CrossRef]
15. Microsoft. VOTT Visual Object Tagging Tool. 2020. Available online: <https://github.com/microsoft/VoTT> (accessed on 30 January 2023).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.