CSCI 1430 Final Project Report:
NFL Position Classification Through YOLO

The Playmakers: John Michael Slezak, Atif Khan, Chenhao Lu, Ruida Zeng.
TA name: Joel Manasseh. Brown University

Abstract

Object classification is a pivotal development in computer vision with applications in medicine, autonomous vehicles, and many more. The purpose of this study is to investigate how context and place within an image can affect a models ability to classify an object. This problem is studied through an analysis of position labeling on a football field where the players have no distinguishing characteristics between them. Thus, the model has to learn from the context and location of the features versus the features themselves to classify them. For these purposes, a custom dataset of NFL formations were gathered and labeled. Then the object detection algorithm YOLO(You Only Look Once) was implemented for training the model. The results varied as the model did a reasonable job classifying wide receivers and cornerbacks. However, it struggled with identifying safeties and the fact that referees were not players. In summary, this model did a reasonable job, but could have been helped by access to more data.

1. Introduction

In this project, we are trying to build an algorithm that can identify the positions of players on an NFL field before the play begins. This problem is difficult because of the similarities between the players when compared with other object classification implementations. All of the players are humans in NFL equipment. Thus, when compared to classifying objects like apples and oranges, context must be taken into account much more than just the features of the object itself. Also, players line up as different positions depending on the play itself. For this project, we classified players based on how they lined up, not their preset position on a depth chart. We will implement YOLO, You Only Look Once, for object identification on a custom dataset created by us using Roboflow labeling software. This type of algorithm would be very helpful for sports analytics, especially, with tracking software once it is started so as to tracks plays and players in real time. More analysis could be done on said plays to both improve the sport and game play. In addition to its application to sports, this progress in moving from feature identification to context and location within an image could help future object detection algorithms in various fields.

2. Related Work

When first researching which computer vision model to train, we came across various options. However, YOLO turned out to be the best option after reading many research papers on it. One of the main strengths of YOLO is that it is extremely fast and makes fewer background errors than traditional R-CNN approaches. [3] Prava et al. Furthermore, the specific version of YOLO we used was YOLOv8. The reason we used this version of YOLO, other than it being a massive improvement from previous versions, is that YOLOv8 was used for UAV aerial photography scenarios which is a similar scenario to NFL plays due to the top-down view of the field. [4] Jiang et al. Lastly, although we needed a fast model that didn't require a lot of time and resources to train, we also didn't want to sacrifice accuracy. Faster R-CNN achieves, one of the most accurate object detection models, achieved a mean accuracy precision score of 76.8 with a Frame Per Second score of 5 to 18. [2] Juan Du. Although it is extremely accurate, it is very slow. Comparatively, YOLO achieved a mean accuracy precision score of 78.6 and a blazing fast Frame Per Second score of 155. [2] Juan Du. And so, with all of this research in mind, we decided to choose YOLOv8 for our model.

3. Method

Before an NFL play begins, all of the players line up in formation on both the offensive and defensive side of the ball. We wanted to find out if we could train a model to be able to recognize what position each player was lined up as. Our approach was to build a custom dataset and train a model on it. We first went through the 2024 Super Bowl, and the AFC and NFC championship for that year documenting every presnap formation. After that we used Roboflow software to label the positions for all 503 images, excluding the offensive and defensive lines. Below is an example after labeling. [1]

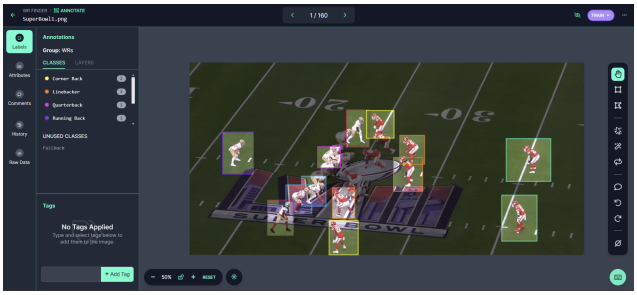

Figure 1. Labeling Example

3.1. Color Matching

The first detection method we tried was direct color matching, and the idea was to use uniquely identifiable colors of each team to detect players (e.g. gold for SF, red for KC), and identify the wide receivers based on the position of the bounding boxes. We first converted the image from RGB to LAB to take into account the luminance factor, and computed the distance between each pixel and the LAB values of gold and red. We then applied thresholding to keep only pixels within an arbitrary distance and converted the image to binary. With OpenCV's blob detection methods, we were able to detect potential player regions and keep the ones larger than an arbitrary size.

In order to know what color to match, we need to know which side is the offensive side. To do that, we utilized the assumption that an offensive formation is usually narrower horizontally than a defensive formation, i.e. players are closer together. For each team, we computed the distance between the x coordinates of the leftmost and rightmost bounding boxes, and the team with a smaller distance will be the offensive team.

To identify wide receivers, we then simply looked for the bounding boxes with the largest and smallest y coordinates, i.e. closer to the horizontal image edges.

3.2. Detection Using YOLO

After accomplishing this on the 503 images from the three games, we exported them in the YOLOv8 format, so as to use that algorithm to do the object detection. The model that we used is called YOLOv8 (You Only Look Once). The way that YOLO works is that it processes an entire image in one go, unlike other systems that analyze parts of the image sequentially. This whole image processing allows YOLO to quickly detect objects. To localize objects precisely, the image is divided into a grid. Each grid section predicts the presence, shape, and size of objects within it. For each detected object, YOLO assigns a confidence score that indicates the certainty of the object's presence and the accuracy of the bounding box that outlines it. To ensure the most accurate results, YOLO employs a technique known as non-max suppression, which refines these bounding boxes by keeping only the most precise predictions and eliminating any overlaps or redundancies. The final output is an image annotated with boxes and labels around each detected object, clearly identifying and locating them within the scene. Figure 2 below shows the architecture of YOLOv8.

In the context of labeling NFL players on the field, as the image of the football field is processed in a single glance, it's divided into a grid, where each section of the grid is tasked with identifying and labeling the players within its bounds. YOLO predicts the presence and exact location of each player, assigning labels such as quarterback, receiver, or lineman based on their positions relative to others and the game's context. Each prediction includes a bounding box around the player with a confidence score that indicates the accuracy of both the player's identification and position. Moreover, using techniques like non-max suppression, YOLO ensures that each player is uniquely identified and labeled only once, even in crowded scenes, resulting in a clear and accurate depiction of all player positions on the field.

4. Results

Figure 3 below shows the results of the color matching algorithm applied on two images from the SF vs. KC super bowl game. The right image is a correct identification of the wide receivers. In the left image, the algorithm failed to identify the offensive team.

4.1. Technical Discussion

Despite successfully identifying the wide receivers, the color matching algorithm has clear limitations that made it impractical to be used. The algorithm is very sensitive to any kinds of background noises, e.g. the NFL logo, end zones, yard numbers, etc. It's also not very generalizable because we need to know exactly what color to use to identify each team. Finally, the algorithm only matches specific parts of the players, instead of the entire player. Depending on the angle and body position of the players, some color features will not be detectable. For example, if the player's jersey is blocked by his arm or another players, the color matching algorithm will no longer be able to detect the jersey. These limitations are the reasons why we switched to a model-based YOLO detection pipeline.

We chose YOLO for identifying positions of NFL players on the field because it has a blend of efficiency, accuracy, and practical applicability which made it suitable for this specific task. YOLO is renowned for its speed, which allows real-time processing and is a significant advantage when working with limited CPU/GPU resources to train a model. Also, YOLO's architecture is designed to look at the entire image during detection, which helps it capture contextual information that is crucial for accurately identifying player positions in the dynamic environments of NFL games. How-

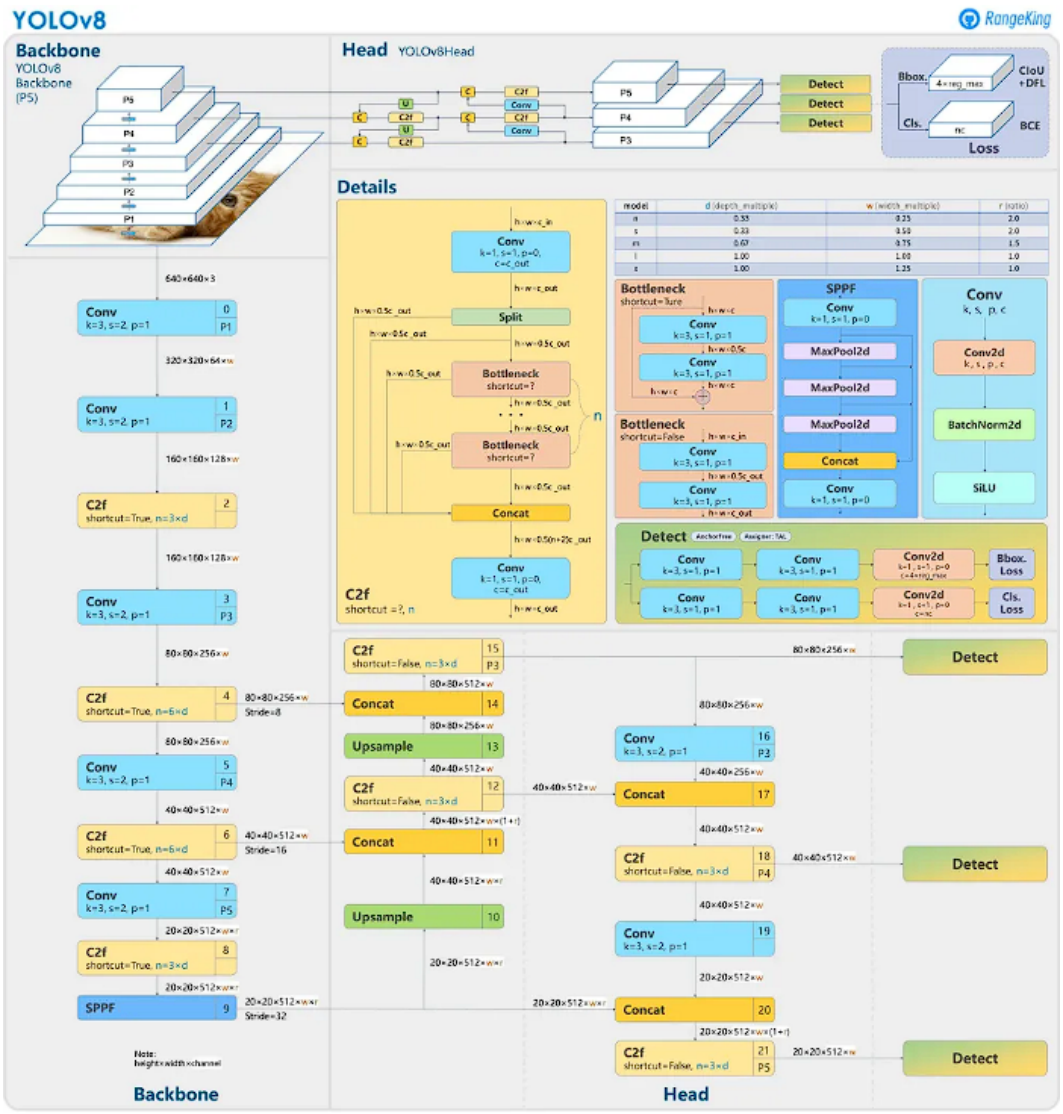

Figure 2. YOLOv8's architecture. source

ever, there are trade-offs to consider. YOLO generally prioritizes speed over precision compared to Transformers-Based Models such as DETR and Swin Transformer, which might result in a slightly lower accuracy, particularly in densely packed scenes typical of sports events where players cluster together. This trade-off means that while YOLO can quickly process images and provide a fast response, it might require fine-tuning and possibly more training data to match the precision offered by other slower but more meticulous models. If we had more time and resources, ideally we would experiment with various other models to see which one gives the best results.

4.2. Testing and Visual

After building the model, we wrote testing and visualization source codes that visualize the results of an object detection model applied to NFL game footage. The code applies the object detection algorithm on loads the detection results, processes the model's outputs, and generates visual representations such as color-coded bounding boxes with labels around detected players. These visualizations display the detected positions of players on the field along with confidence scores.

As shown in Figure 4, our model is performing quite well in these two test cases on the real-time NFL game footage, where it detects and classifies players by their positions with varying confidence levels. In the first image, the model identifies one of the wide receivers with confidence scores of 0.83 correctly, and one of the linebackers detected with scores of 0.83 as well. The second image presents a similar scenario.

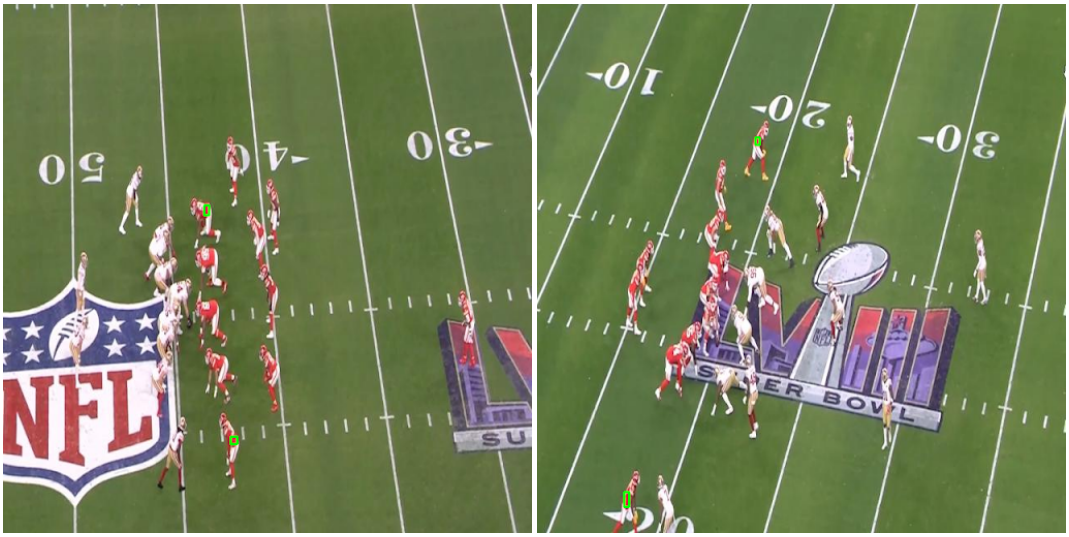

Figure 3. Examples of the wide receivers detected using color matching.

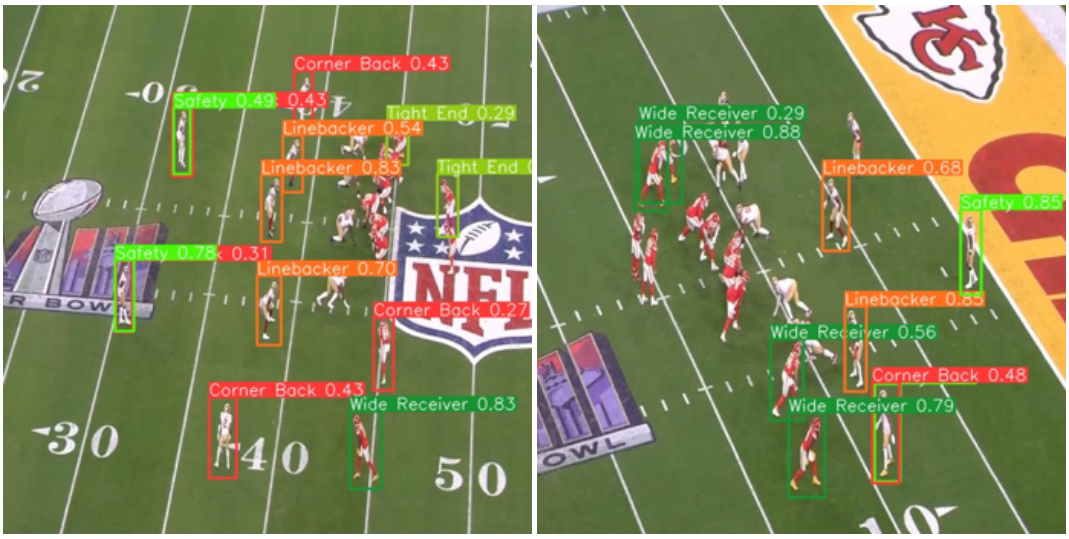

Figure 4. Examples of the (mostly) correct matches.

4.3. Results Analysis

Figure 5 illustrate various performance metrics of the object detection model. The first image, the confusion matrix, which highlights the number of true positives, false positives, and misclassifications for each player position. For example, the model shows strong performance for detecting wide receivers and cornerbacks (which is what we desired in the first place) but struggles with fullbacks and other positions.

The second image, the F1-Confidence Curve, shows the F1 score against the confidence levels for each position, with an overall F1 score of 0.62 at a confidence level of 0.229.

Lastly, the Precision-Recall Curve, indicates the precision and recall values for different player positions, with a mean average precision (mAP) of 0.759 for all classes at a 0.5 threshold.

4.4. Future Works

In reviewing the results, we identified several areas that could be improved upon in future studies. As shown in Figure 6, this includes correcting logical inconsistencies observed, such as the misclassification of a safety positioned on the offensive side or the identification of multiple quarterbacks in a single play. These issues highlight potential areas for refining the accuracy of our model. For example, addressing the misclassification errors would involve fine-tuning the model to better understand the contextual positioning of players on the field, and that there is usually a clean boundary separating the offense and the defense. Moreover, expanding the training dataset to include a wider variety of game situations and team formations could also contribute to a more robust and generalizable model, since the players with the same

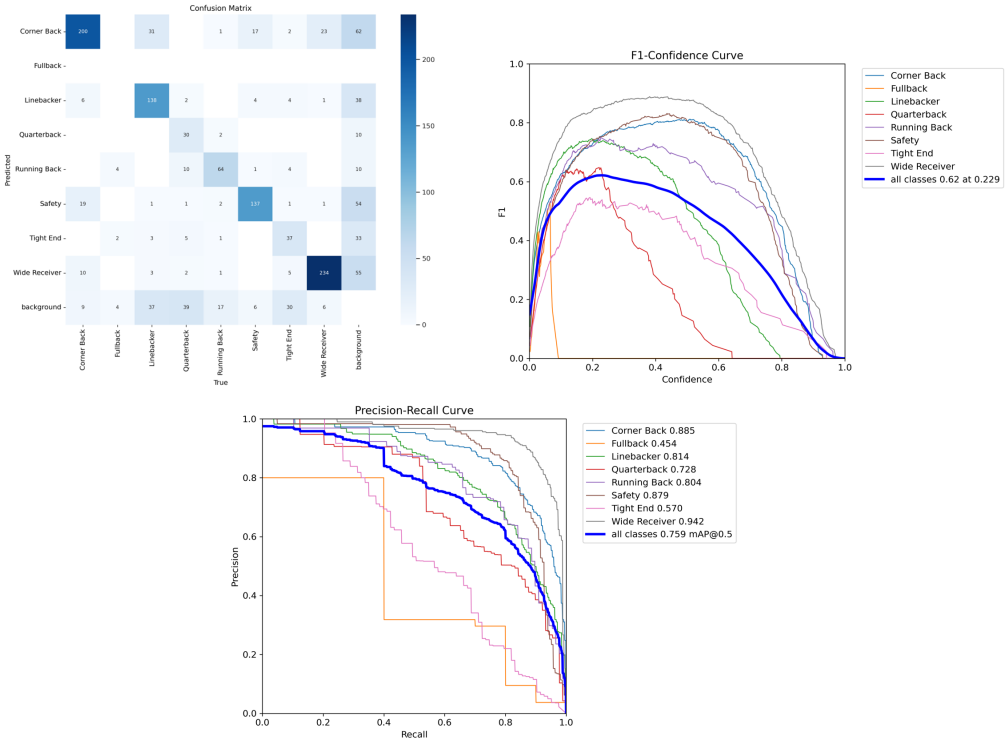

Figure 5. Testing Result with confusion matrix, confidence curve, and precision-recall curve.

roles are often positioned differently on different plays.

5. Conclusion

In the end, we used the YOLO model on a custom dataset to create a model that would identify the positions of players on an NFL field. The ability to perform this analysis in real-time has many applications for the sports world and beyond. For example, being able to identify players positions and then potentially tracking their movements could help with performance analytics for both the player and the plays themselves. In addition, access to more data in sports over the years, has helped develop new techniques and outlaw plays that were particularly harmful to players safety. The ability to track how formations affect injury risk could open up another pathway to help players remain healthy. Lastly, we found that how YOLO worked helped it analyze the full context of the image allowing for it to better identify the players. Designing and building models that take into account context will help with many future object classification algorithms regardless of the use case. The entire sports and media industry is a massive component of the world economy. Improving the games and keeping athletes healthy while they play and after they retire is a huge endeavor that needs every tool at its disposal. This position classification model will help to achieve that goal.

References

[1] *Roboflow*. 1

[2] Juan Du. Understanding of object detection based on cnn family and yolo. *Journal of Physics: Conference Series*, 1004(1):012029, apr 2018. 1

[3] Jana, Arka Prava, Biswas Abhiraj, and Mohana. Yolo based detection and classification of objects in video records. *2018 3rd IEEE International Conference on Recent Trends in Electronics and Information and Communication Technology*, 2018. 1

[4] Peiyuan Jiang, Daji Ergu, Fangyao Liu, Ying Cai, and Bo Ma. A small-object-detection model based on improved yolov8 for uav aerial photography scenarios. *Sensors*, 2023. 1

Appendix

Team contributions

Please describe in one paragraph per team member what each of you contributed to the project.

John Michael Slezak Created the dataset and labeled all of the images that the YOLO model was eventually trained on. Helped to write the code to train the YOLO model. Contributed to the results and methods section of the poster and the methods, introduction, and conclusion section of the report.

Atif Khan Worked on writing the code and training the model for the player detection algorithm as well as

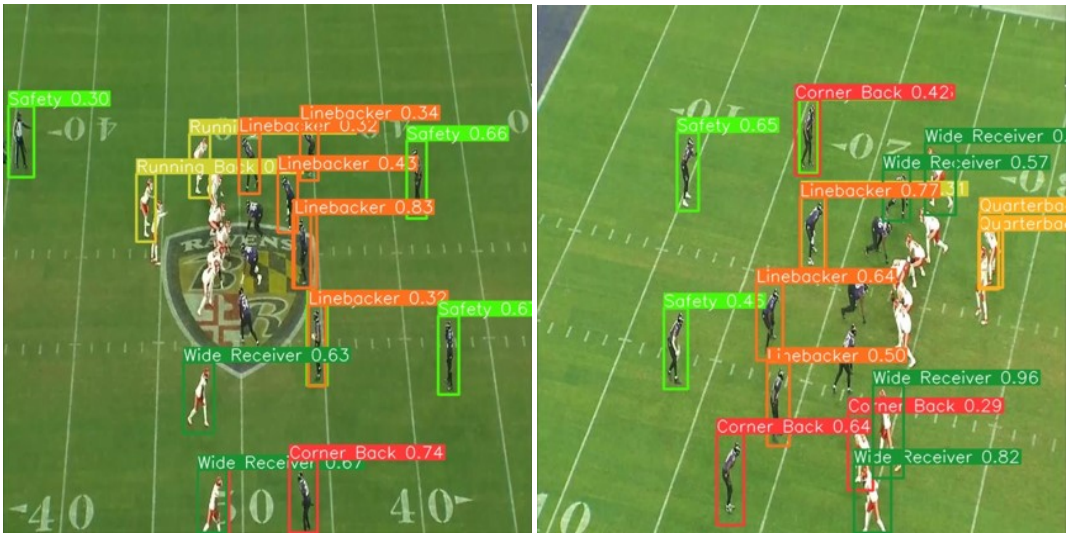

Figure 6. Examples of the incorrect matches that defy logical expectations.

researching the YOLO model. Contributed to the methods, technical discussion, and related works section of the report as well as the beginning sections of the poster.

Chenhao Lu Researched methods for object detection, wrote and tested the code for applying the pre-trained YOLO model without fine-tuning, wrote and tested the code for the color matching algorithm.

Ruida Zeng Wrote a script that computes a customized filter for every image and then applied them in order to improve the color matching algorithm. Additionally, utilized the YOLO model and weights on a brand-new dataset containing NFL game footage to further test the accuracy of our object detection model. Contributed to the testing, results, and analysis sections of the report, identifying areas for future improvement.