CSCI 370 Final Report

Snap & Detect

Ava Courtney
Jackson Wray
Jakobi Wells
Tommy Jernigan

Revised December 9, 2025

CSCI 370 Fall 2025

Prof. Kathleen Kelly

Table 1: Revision history

Revision	Date	Comments
New	August 27th, 2025	Completed sections I-V.
Rev – 2	September 10th, 2025	Quantifying some requirements and adding team profiles, references, and key terms.
Rev – 3	September 18th, 2025	Completed sections VI-VIII.
Rev – 4	September 21st, 2025	Filling out individual profiles.
Rev – 5	October 9th, 2025	Updating Ethical Considerations section to match assignment guidelines.
Rev – 6	November 8th, 2025	Revising sections I-V to have more depth.
Rev – 7	December 1st, 2025	Commenting suggested changes after peer review.
Rev – 8	December 8th, 2025	Inserting suggested changes before final submission.
Rev – 9	December 9th, 2025	Updating visualizations and finalizing edits.

Table of Contents

I. Introduction

In many college football programs, film analysis plays an essential role in a team's success. By effectively preparing for offensive plays, the defensive players have additional knowledge to cover and pick offensive players strategically. To achieve this, coaches rely heavily on reviewing opposing team's footage of previous games to gain an understanding of how they handle different situations. By labeling these formations and positionings, analysis is run on the most frequent routes, runs, and passes made. However, this labeling requires an extensive focus and time investment by the coaches. This leaves them with less time to coach their players. Furthermore, since the labeling process is manual and lengthy, it is prone to human error and misinterpretation.

Our team is collaborating with Coach Anthony Makransky to develop a computer vision system that automates the labor-intensive process of labeling football film. Currently, coaches use an application called Hudl to manually tag positions, formations, and plays. Many of the features Hudl offers help coaches store and share data easily. While these features aid in organizing the process with several coaches efficiently, there are few tools to aid in the labor of the labeling process. By leveraging computer vision techniques, our system aims to streamline the labor, improve efficiency, and reduce the repetitive part of the workload on coaching staff.

There are two key components of a system that can automate this functionality. First, there needs to be a way to accurately track and classify the players on the field. Next, analysis should be run on the identified players and their locations to determine the offensive formation and plays. This report is primarily concerned with getting the first step working, with some discussion of any experimental work into the second step.

II. Functional Requirements

The system our team produces must provide the following core functionalities in order to be beneficial to our client:

1. Play Ingestion
    - Accept an input video consisting of a trimmed football play.
    - Automatically detect the snap, selecting the prior frame for future analysis to be run on.
    - Ingest multiple plays automatically when given a folder for batch processing.

2. Reliable Normalization of the Field
    - To handle the multitude of angles and positions that the film can be provided with, a homography needs to be performed to transform the players into a consistent birds eye view.
    - A minimum of four keypoints per frame are needed to compute this homography.
    - These keypoints need to be identified with a minimum of 95% accuracy.

3. Player and Position Detection
    - Identify all visible players on the field with 95% accuracy.
    - Classify the position of players on the field with 85% accuracy.
    - Position will be classified from the following set {qb, oline, tight_end, running_back, wide_receiver}.

III. Non-Functional Requirements

The functional requirements above must be supported by the following constraints and quality standards:

1. Performance
    - Process a single play clip within 1 minute and 30 seconds on standard team hardware. This involves the identification of the snap, the identification of key points as well as performing the homography transform, and the detection of offensive players as well as their positions.
    - This requires efficient use of the CPU and GPU. Multi-threading could prove a useful way to utilize all available resources in the future when formation classification is implemented.

2. Accuracy
    - The following requirements are benchmarked from a test and validation suite using at least 10% and 20% of the dataset respectively.
        i. Achieve at least 95% accuracy for field marker detection to identify the minimum of four key points needed for our homography transform.
        ii. Achieve at least 95% accuracy for the player detection.
        iii. Achieve at least 85% accuracy for the classification of player's position.

3. Usability
    - Once a video clip is selected, the pipeline requires no user interaction until the pipeline has finished running and all the players before the snap have been identified and their locations transformed to the bird's eye view.
    - Results should be presented in a clear and readable manner to allow for quick visual validation of our systems predictions.
    - Our application's UX should be intuitive for someone familiar with the Hudl web application.
    - Any failures during processing should be displayed in an easy to understand manner during a single processing run. During a batch processing run, any failures should be logged as to not stall the concurrent jobs.

4. Reliability
    - Handle varied camera elevation angles, lighting conditions, and video quality without errors.

- Our system does not handle film where our view to the yard markers is obstructed by things such as snow or significant paint weathering.

5. Maintainability
    - Code is modular and well-documented so future teams can retrain models or update labeling logic.
    - We will define clear separated components for each identifiable step in our pipeline, so that errors are easily traceable.
    - All code and configuration files are clearly documented, including setup instructions, model architectures, and parameter explanations.
    - We provide clear and easy to understand setup instructions so that future developers have a smooth onboarding.

6. Scalability
    - The system is capable of handling single play clips which should be split out from the whole game film via Hudl.
    - The system supports queued or parallelized batch operations to handle large video collections efficiently.
    - If better models are trained in the future, our codebase is structured for easily switching the model used during early detection steps.

IV. Risks

The development of a multifaceted football play detection system involves several inherent risks that can impact the accuracy, reliability, and timely completion of our project. These risks are divided among technical and skill-related categories.

Technical Risks

- Incorrect or unreliable labeling of player positions will lead to errors in the future formation classification which could result in misleading statistics. These inaccuracies can stem from limitations in the model's training data, congested player arrangement, or visual confusion between similar positions such as tight end and offensive line.
- Because pre-trained models are not available, we need to collect and label training data, which is a time-consuming and resource-intensive process. Additionally, there is the possibility for inconsistencies in labeling data for the player's position since a human's estimation is subjective for easily confused positions.
- Training deep learning models requires significant computational power, which may exceed the team's available hardware or allocated Roboflow training tokens.
- The system's effectiveness is highly dependent on the consistency and quality of input video data. College football footage varies dramatically across teams and locations; differences in camera angles, lighting, and pixel resolutions can significantly affect model accuracy. For example, poor lighting or glare obscures key visual features, while extreme zoom levels or unconventional camera positions distort spatial relationships between players.

Skills Risks

- Only two team members have prior experience in computer vision, which may slow development and debugging.
- Only one member has extensive knowledge of football plays and formations. Additional learning or consultation with the client is required to ensure correct labeling.
- Given the limited development window of field session and the extensive development needed for our project, there is a significant risk that there is not sufficient time to master the technical skills needed to achieve our definition of done. This results in underdeveloped or invalidated components in our pipeline. If this becomes the

case, our team will take the time needed to thoroughly document these components to help quicken the learning curve of future developers. Additionally, we are thorough in our communication with our client, informing him of any components that are unfinished.

Collectively, these risks highlight the project’s dependence on high-quality data, adequate computational resources, and specialized expertise across both technical and domain areas. While each risk presents unique challenges, their combined effect underscores the importance of rigorous planning, clear communication, and adaptability throughout the development process. Recognizing these risks early in the project lifecycle allows for our team to anticipate obstacles and maintain realistic expectations regarding the system’s achievable performance and scope.

V. Definition of Done

The project will be considered complete when the following criteria are met:

1. Minimal Useful Feature Set
    - **Video Ingestion:** The system properly takes a video of a single football play and accurately selects a frame before the snap to run analysis on.
    - **Data Generation:** Correctly identifies yard markers and player’s field placement to accurately transform data to a normalized view.
    - **Player Classification:** Correctly classifies the position of each player on the field.
    - **Ease of Use:** Shallow learning curve for non-technical users interacting with our application.
    - **Maintainability**: Our project is expected to be handed off to future developers, so it is vital that we leave accurate, thorough, and digestible documentation of our codebase.

2. Client Acceptance Tests
    - **Validation:** Coaches validate results using a set of pre-labeled clips not included in training.
    - **Performance Comparison:** Success is measured by comparing automated labels against manual labels.
    - **User Feedback:** The client interacts with our application, and we resolve any design based improvements he suggests.
    - **Assessment:** The client will confirm that the tool accurately identifies and translates the view of all players.

3. Delivery
    - **Executable Distribution:** Delivered as a packaged executable with clear usage instructions. Since our client isn’t technical, our application launches similar to other desktop applications he is used to.
    - **Documentation:** Our codebase is shared using Github. We will also include documentation detailing setup, dependencies, and instructions for retraining the model if needed, or for future field session groups.

VI. System Architecture

Technical Design Issues

1. Model Training for Keypoint Detection
   Our plan relies on accurate keypoint (yard markers) detection to enable the perspective transform. However, we are encountering challenges in training models to select these keypoints reliably. Until we resolve this issue, we will block the future implementation of any formation labeling.

2. Hudl API Limitations
   We discovered that direct integration with Hudl's API is not possible. As a result, we cannot support features like authenticating with Hudl, automatically re-uploading our labeled video, or downloading directly from Hudl for our application. The workaround will be to have the client manually download video files and then receive output as a CSV file, which can be reuploaded to Hudl.

3. Computer Vision Limitations in Football
   Football is inherently a sport where players make contact frequently. When players bunch up in one location, this could cause layers of players where one player blocks the view of another player from the camera. This may cause issues with identifying players properly.

4. Neural Network Limitations
   Even Neural Networks with a 99.9% accuracy are wrong 0.1% of the time. To address this, we would like to add functionality for the client to correct mistakes and know where they are likely to occur.

System Design

Our developed system has two main components: AI analysis of footage and a GUI to interface with the AI and ensure accuracy.

Our AI system has 3 distinct steps. We use a trained YOLO model to determine player locations relative to the camera that we use for snap detection, a different trained YOLO model to identify yard markers, and another trained model to determine player's position such as quarterback. YOLO is the current industry standard for computer vision models. It leverages its accuracy and speedy processing time to become quite useful for real-time applications. While our application does not run in real time, the large amount of processing required must be done efficiently to ensure our product is reliable and time saving.

Using the player location data, we identify the time of least movement, which is the most likely time of the snap. We then use our player position detection on this frame. We combine the player position and yard marker data to create a homographic image of the field. With different camera elevation angles, there is a distorted view of the field. Using homography, we generate a bird's eye view of the field regardless of this angle. Homography is a mathematical operation in which you identify "key points", or known positions on a field or court (yard markers in our case), to transform a side camera image to reflect the true positioning on a field or court. This allows us to have normalized locations of players across different camera elevation angles.

For the ease of use of our client, we built a GUI that is very similar to Hudl's to minimize the time it takes to learn to use our software. The Hudl GUI which we are modeling after has 4 main panels. This includes video playback of the currently selected folder, folder access to select and view different available sets of footage, and a data sheet to gather statistics and labels about the footage. This is arranged with video playback in the top left, file access in the top right, and the data sheet on the bottom.

However, we made additions to the Hudl interface to better suit our client's needs. One feature we added to the video panel is the option to display bounding boxes for players and yard markers. This will allow the client to see if any players are not being identified and if any yard markers are being mislabeled. In addition to the bounding box toggles, we added a legend that can be referenced to give clarity on which color bounding box corresponds to which position. By doing this, our client can perform a quick visual inspection to verify the classification of our players. An additional panel we added is a "virtual field." This will be placed in the bottom right. The purpose of this is to give a view of where our computer vision pipeline has identified the players to be on the field. This can provide confidence to the client that the data being used to predict formations and other labels is accurate. AI systems are inherently imperfect; if the client can identify that the position data generated is not useful, they can correct the system or label the play clip themselves to ensure no false statistics are being generated before strategizing. To interface with our AI pipeline, we added buttons to the data sheet to process the current clip and/or process the current folder.

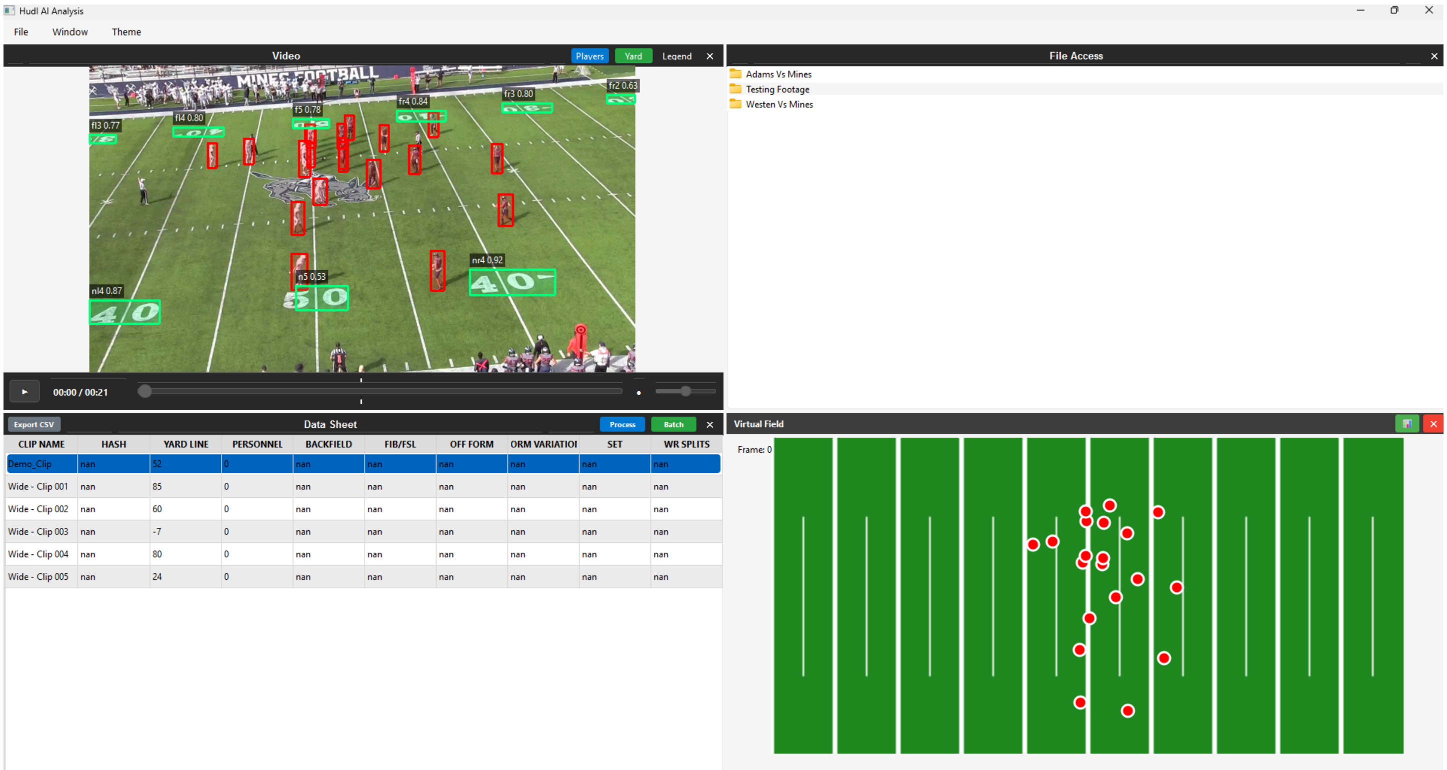

Figure 1: Hudl-like User Interface

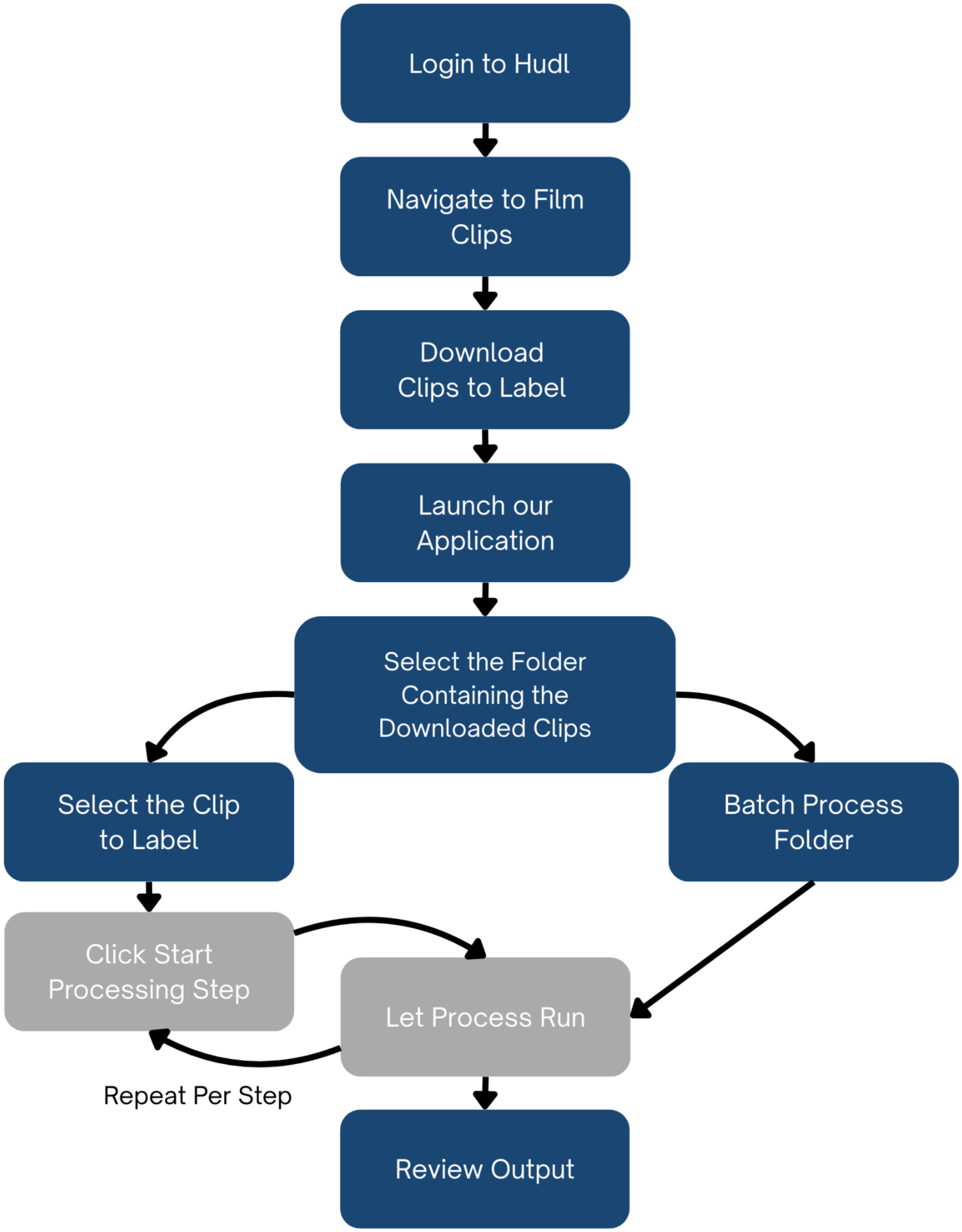

Figure 2: Client Use

The above figure demonstrates the flow that our client follows while using our product. This involves a manual authentication to Hudl to select which film he wishes to label. We would prefer to automate this process as well, but as we have limited access to the Hudl API, this manual process is something which we will require. However, this sacrifice is not a large one, as our client is already familiar with the Hudl interface as well as navigating the library feature of Hudl.

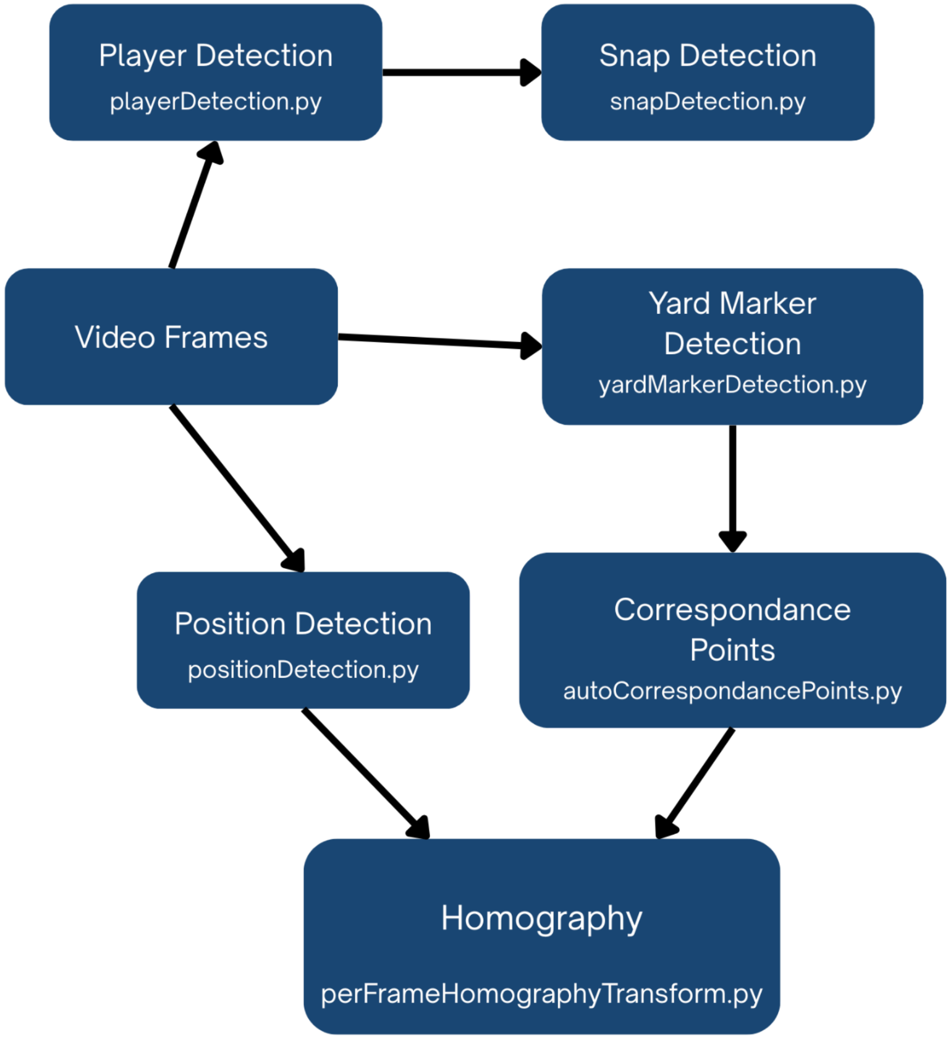

Figure 3: AI Pipeline

The above figure demonstrates a more in-depth execution of our pipeline flow. This is contained within the grey steps of the *Client Use* diagram. We are using Python files for many of our different AI applications in this product, each of which are labeled above. This pipeline can run with or without human interaction depending on batch or single process.

The future part of our AI system will involve inputting this player location and classification data into two trained Convolutional Neural Networks (CNNs). These CNNs will use labeled data provided by the client to learn different formations and aspects of a play that the client has requested. One CNN will process the static image before the snap to analyze the beginning formation. Label accuracy for complicated aspects such as the formation will likely benefit from the use of a CNN, while other aspects such as the yard line and hash mark of the play could be identified with empirical methods. The second CNN will be used to process all positioning data after the snap. This will detect aspects about how the play unfolds, such as which receiver the ball was thrown to, or if the play was a pass or run. To help normalize inputs, we will base our origin for the coordinate system at the center of the line of scrimmage. This will help the model learn more efficiently, as it will have a localized point for each play to base its calculations on.

VII. Software Test and Quality

To ensure software quality, we will have several code reviews and refactors to ensure interpretability and efficiency.

Test	Description	Results
Player Detection	We would like to see a 95% accurate detection of players. This can be tested by iterations of cross-validation.	97% Accuracy
Position Classification	We would like to see an 85% accurate classification of the players assigned positions. With all defensive players labeled as defense, and offensive players labeled accurately from the classes of [qb, running_back, tight_end, oline, wide_receiver]	89% Accuracy
Keypoint Detection	We would like to see a 95% accurate detection of keypoints. This can be tested by iterations of cross-validation.	96% Accuracy
UI Functionality	The UI should behave very similarly to Hudl. We will communicate with the client to ensure the most important features are preserved.	UI is functional to the client's liking
UI Efficiency	The UI should be efficient and reactive to button presses. It also behaves similarly on Mac, Windows, and Linux systems.	UI is not as efficient as we'd like on less powerful computers; however, it functions properly on powerful computers. This is naturally a very computationally intensive task and our client has access to proper hardware, so this is satisfactory.
Formation Classification Accuracy	We would like to see a 95% accuracy across all models used to label categories. This can be tested by iterations of cross-validation.	Not yet implemented
Formation Classification Efficiency	We would like efficient processing of the footage and computation. We have not gotten a baseline for our pipeline, but we will further optimize from our initial baseline.	Not yet implemented
Homography Accuracy	While difficult to verify numerically, we can visually confirm if the	After several improvements of keypoint detection, our homography

	transform is accurate by reviewing the transformed footage.	is satisfactory. There are still some errors, so additional data for model training would likely lead to benefits.

VIII. Project Ethical Considerations

The first ethical consideration of our project involves our protection and proper use of the Colorado School of Mines Football film. We must be responsible in our use of these clips, this entails preventing any leaking of the clips or sharing of the knowledge gained through the clips. For this reason, our use of Roboflow to label datasets for our custom models is within a private project on Roboflow. Failure to properly handle the clips given to us by our client violates our confidentiality agreement and violates ACM principles 1.6 and 1.7, which are “respect privacy” and “honor confidentiality”. These principles apply directly to our agreement with our client, and it is something that we take great care to achieve.

The second ethical consideration of our project is related to an honest representation of our product to our client. If it becomes the case that our model or system is improperly labeling the film, we could harm the performance of the Colorado School of Mines Football Team by hindering their understanding of opposing teams' offensive formations. For this reason, it is our ethical responsibility to convey the unreliability of computer vision models and other forms of AI processing to our clients. Additionally, future development of formation detection systems will help aid our client in understanding the results of the labeled clips through color-coded result spreadsheets, as well as validating the results of our models by comparing its results to previously manually labeled data. These solutions will help ensure that we do not violate IEEE principle 3: “Software engineers shall ensure that their products and related modifications meet the highest professional standards possible”, as well as ACM principle 2.5: “ Give comprehensive and thorough evaluations of computer systems and their impacts, including analysis of possible risks”. While we may not have the time or resources to produce AI models that meet the highest professional standards possible, we will ensure that we have professional level documentation, communications, and usability for our client. Furthermore, as discussed above, we will be fully transparent in our demonstrations of our product to the client. We place great consideration in ensuring that our client understands all of the risks and limitations that our product has.

Michael Davis’s Tests

1. **Harm Test:** We must consider whether our use of AI labeling does less harm than alternative methods. Our system aims to save time and improve efficiency, but if it produces unreliable outputs, the resulting misinformation could outweigh the benefits. Therefore, thorough validation and testing are necessary to minimize harm.
2. **Reversibility Test:** If we were in the client’s position, we would want complete transparency regarding the accuracy and reliability of the system. This means we will always communicate limitations clearly and avoid overstating the system’s capabilities. Taking this perspective helps ensure fairness and builds trust in our partnership.

In regards to our software plan, a failure in any of our requirements will be communicated to the client clearly to ensure knowledge of the risks of our software. For example, if we had failed to meet our goal of 95% accuracy for yard marker detection, we would communicate this with our client so they are aware of how these results may not always be correct. Another example is our UI functionality. While we are designing our UI visually and functionally similar to Hudl, there are going to be differences. We planned a session where we taught our client how to fully utilize our program and emphasized the differences from Hudl. This ensures that the client is comfortable with their user experience and we could make any changes they believed would elevate that experience.

Security concerns of user data should always be taken seriously. In our case, the most user data required is footage to be imported to the program. This football film is protected and should be stored securely. As our program does not interact with the internet, this alleviates most concerns.

IX. Project Completion Status

This project's main purpose is to create a tool that can view football film and automatically generate appropriate labels to classify the formation or play. While it is easy to conceptualize a pipeline to process this, the implementation is much harder. Currently, the player detection does a good job at differentiating referees and the two teams, but we've had issues with the yard marker detection. There are still many cases where it fails to identify yard markers or misclassifies. This leads to a miscalculated homography, so the player position data is useless to classify plays. We have been able to greatly improve this with additional training data throughout the semester, but more is likely needed. Additionally, it should be noted that our position classification only works reliably for the few frames before the snap. After this point, its accuracy significantly decreases. Any future implementations that involve analysis after the snap will need to further fine tune our model to handle position classifications later in the play.

We were able to give our client a successful deliverable. We have built out a UI that the client finds intuitive and is very excited about. This features a functional folder and video selection, video playback with bounding box overlays, a virtual field to display player detection, and the ability to launch our processing pipeline, although the pipeline itself is not complete with formation detection and labeling. This provides a great stepping stone for future projects to improve the pipeline and provide intentional functionality.

Ultimately, our client's main goal was to save him time. At the current state of our project, we have not completed the second half of our AI pipeline to analyze the data we generated. However, we were able to successfully set the foundation for a very versatile formation prediction pipeline that can be developed in coming semesters. One of the first conversations we had with our client involved a reframing of his expectations from just detecting the quarterbacks positioning to getting an accurate and adaptable estimation of all players locations. We identified that training a model to detect hyper-specific tasks would be a waste of time in the long run. Thanks to these conversations, we were able to take our client's idea and help him to begin what will be a very impressive and large scale project.

X. Future Work

To reach our initial goal of saving our client time, there are only a few things left to implement

- Increase yard marker detection and player classification accuracy, precision, and recall to satisfactory levels.
  - Improve Yolo models with more data
  - Test different training methods and data augmentation
- Train Convolutional Neural Networks to intake player positioning data to classify.
  - Transform player data to be labeled
  - Create tensors for player data
  - Label positions and formations
  - Determine thresholds and confidence for predictions
- Include the ability for our client to correct any incorrect detections.
  - For example: If there is no yard marker detection in a frame and the homography cannot be computed, the client should have the ability to manually identify the yard markers, and the rest of the pipeline should run
  - These corrections should be outputted and used as data to fine tune our models with
- Interface with the Hudl API to pull footage from the cloud automatically without downloading separately
  - Attempt to find commercial-free Hudl API calls
  - Create a Syncing System between Hudl and our program

- Create a dashboard to show trends and insights into how the opponent is playing.
    - Possible statistical analysis on data; depends on client's needs
    - Potential graphical methods to display information

XI. Lessons Learned

There were several lessons we learned while developing this project.

Technical:

- Computer vision models are not perfect. This project relied on accurately identifying real-life points on the field to transform player positions to a non-distorted view induced by the camera angle. Without this accurate detection, we cannot perform our analysis. We found the labeling process to be extremely time intensive, limiting our ability to reach our goals.
- Pipelines are only as good as their components. Due to the inaccuracy of the yard detection, this limits our ability to generate data for training a CNN to classify plays, and would certainly lead to misclassifications if we had a trained CNN. However, we found that more data and experiments with augmentation improved accuracy, so dedicating more time to labeling can alleviate this issue.
- Camera quality is vital to efficient computer vision techniques. Higher-quality footage is directly related to higher-quality performance on all fronts.
- Computer vision models and CNN networks can become incredibly expensive to train. This resulted in a fair bit of expenses for this project and has led to significant delays at times as we wait on models to train.

Professional:

- Consistent conversations with our client was vital to our efficacy as a team. When we found ourselves stuck and unsure of how to tackle certain issues, our client was able to let us know where he next wanted our efforts placed. This helped us to realign on issues we may have set aside or not considered.
- Plans can change. This is one of the reasons that getting a clear definition of the project from the client early on is so important. This definition may see modifications or sometimes be scrapped all together, but by having clear conversations and putting effort into getting specific requirements, the stress of major changes can be alleviated, and developers can understand where the changes come from.

Team Based:

- Establishing an easy and accessible form of communication is very important to our success as a team. We chose to use both a discord channel and text message group chat. As the semester began to pick up, we started to stray away from the discord channel and lean more heavily into the text thread. This can be attributed to the group chat's accessibility. Any quick ideas or updates we had could be easily seen and had on our phones, opposed to needing our computers for the discord messaging.

XII. Acknowledgments

Thank you to our client Anthony Makransky for making the time to meet with our team consistently throughout the semester. Consistent and clear communication with him was extremely beneficial to our development process. Through this, he provided guidance and reframing when we found ourselves blocked by model performance. We also wish to express our gratitude to Kathleen Kelly for allowing us to be a part of such an interesting project.

Additionally, we enjoyed having Max Sobell as our advisor this semester. He kept us on track in our sprint meetings and gave us applicable feedback throughout the semester. He was also very encouraging when discussing our progress and helped us to understand what was expected of us.

XIII. Team Profile

Ava Courtney

- Senior - General Computer Science B.S.
- Some computer vision experience.
- Enjoyed combining multiple applications of computer vision into a single pipeline.

Jackson Wray

- Senior - M.S. Data Science, B.S. Computer Science
- Experience with AI concepts.
- Enjoyed learning more about computer vision, UI design, and AI implementations.

Jakobi Wells

- Senior - B.S. Computer Science, B.S. Applied Mathematics & Statistics
- Basic experience with machine learning and AI implementations.
- Enjoyed building a full software solution and engaging with computer vision/ machine learning approaches.

Tommy Jernigan

- Senior - B.S. Computer Science, Minor in Computational and Applied Mathematics
- Computer Vision and basic AI experience.
- Enjoyed learning more about computer vision and AI concepts.

Appendix A – Key Terms

Term	Definition
Hudl	*The video analysis platform used by coaches to review, tag, and share plays.*
Trimmed play	*A video clip that contains just one football play, cut down from the full game film.*
Accuracy	*The percentage of correct predictions made by the system compared to the actual ground truth.*
Formation	*The arrangement of offensive players on the field before the play starts.*
Homography	*Linear Algebra/ Computer Vision process for transforming one perspective to another.*
Correspondance Points	*Identifiable points on an image that can be directly translated to a real-world location.*
You-Only-Look-Once(YOLO)	*A real-time computer vision tool used for detecting, classifying, and tracking objects in digital media.*
standard team hardware	*HP ENVY Desktop TE02-0xxx*