Faculty of Mathematics and Information Sciences

WARSAW UNIVERSITY OF TECHNOLOGY

Group Project - LAB 1

Project of an application for Analysis of Football using Computer Vision

Osman Aliyev
Hubert Jaczyński
Aleksander Klepka

October 2024

Table of Contents

Abstract

The aim of this work is to develop an application that leverages computer vision to analyze football match footage and extract comprehensive statistics about the players and the match overall. This project involves implementing a computer vision model to accurately identify and track key components in the footage, including players, balls, goalposts, and referees. With these components recognized, the next focus will be on computing detailed match statistics and relevant metrics. The main objective is to provide valuable insights into football matches, enhancing both player performance analysis and game strategy evaluation. To achieve proper analysis, the DFL - Bundesliga Data Shootout dataset will be used, which consists of 460, 30 second videos in mp4 format from various football matches. The implementation will be carried out in Python, with the use of OpenCV, YOLO and Roboflow.

1 History of Changes

History of Changes			
Version	**Date**	**Author**	**Description**
01	18/10/2024	Osman Aliyev Hubert Jaczyński Aleksander Klepka	Abstract, Vocabulary, Specification, Project Schedule and Risk Analysis

Table 1: History of Changes to the project

2 Vocabulary

You Only Look Once (YOLO) - Real-time object detection algorithm that frames detection as a single regression problem, predicting bounding boxes and class probabilities directly from full images. YOLO is known for its speed and accuracy, making it a popular choice for object detection tasks [1].

Computer Vision (CV) - The study of digital image structures and patterns, which is a layer of image analysis above that of image processing and photonics. Computer vision includes image processing and photonics in its bag of tricks in its pursuit of image geometry and image region patterns [2].

Deep Learning - sub-field of machine learning that involves the use of large neural networks to create models which are able to make accurate data-driven decisions. Suitable for creating models based on complex and substantial in size datasets. As such deep learning models are widely used in modern technologies like smartphones and self-driving cars as well as healthcare [4].

Convolutional Neural Network (CNN) - type of neural network designed to process grid-like data, such as images. It uses the mathematical operation convolution to automatically detect features like edges and textures. This makes CNNs highly effective for tasks like image recognition by preserving spatial relationships between pixels. They have been widely successful in computer vision applications [5].

OpenCV - Open-source computer vision library designed for real-time applications. It provides a wide range of tools for image processing, including object detection, image recognition, and image segmentation. OpenCV is written in C and C++ and has bindings for Python, making it a popular choice for computer vision projects [6].

Roboflow - Tool that simplifies the process of training computer vision models. It allows users to upload, annotate, and preprocess images, before exporting them in a format suitable for training. Roboflow supports a wide range of model architectures and frameworks, making it a versatile tool for computer vision projects. [7]

DFL - Bundesliga Data Shootout Dataset - Dataset created as part of the Bundesliga Data Shootout competiotion, organized by DFL in partnerhip with Amazon Web Services and Sportec Solutions AG. It consists of video clips from Bundesliga football matches, with the goal of providing a dataset for the development of innovative data-driven solutions for football analysis. [3]

Football Data from Transfermarkt Dataset - A clean and structured dataset containing football data from Transfermarkt, which is a well-known website featuring football statistics, market values, and transfers. It consists of 60,000+ games from various seasons across all major competitions, 400+ clubs from those competitions, 30,000+ players from those clubs, 400,000+ historical records of player market valuations, and 1,200,000+ player appearance records from all games.[8]

3 Specification

3.1 Executive Summary

The main goals of this application are to analyze football clips using a deep learning model to provide users with insightful football analysis, and to enable users to search for football players and access a dashboard displaying detailed information and statistics about the specified player. Additionally, the dashboard offers filtering options, allowing users to refine the data presented and gain more in-depth insights.

3.2 Functional Requirements

As a user, I would like to:

- Select a specific clip from a predefined list of analyzed data. (Must have)
- Upload a football clip and send it for analysis so that the model can process the video. (Could have)
- View the insights after the analysis of the football clip so that I can understand the key findings and insights. (Must have)
- Search for a football player so that a dashboard displaying the player's information is shown. (Must have)
- Filter the data in the dashboard by specific criteria so that I can view relevant information tailored to my needs. (Must have)

Id	Actor	Name	Description	System Response
User	User	Static List of Analyzed Clips	Displays a static list of previously analyzed video clips for the user. The user can view, filter, or select from the list.	The system shall present a static list of analyzed clips, allowing users to filter by criteria such as date or category.
		Uploading Videos	Allows users to upload video files for analysis.	The system shall enable users to upload videos in various formats such as MP4, MOV, etc.
		Analysis of the uploaded Clips	Performs automatic analysis of the uploaded clips and presents results such as key moments, player performance, etc.	The system shall analyze the uploaded clips using predefined algorithms and present key results based on player performance and game statistics.
		Player Statistics Dashboard	Presents a dashboard which displays the statistics and information about the user-selected player.	The system shall refer to Transfermarkt data and display it in a dashboard.
		Filter Dashboard	Provides filtering options for dashboard data, allowing users to refine their search by criteria such as player name, player team, and date range.	The system shall allow users to filter dashboard data by various criteria such as player name, player team, and date range.

Table 2: Description of the Use Cases diagram

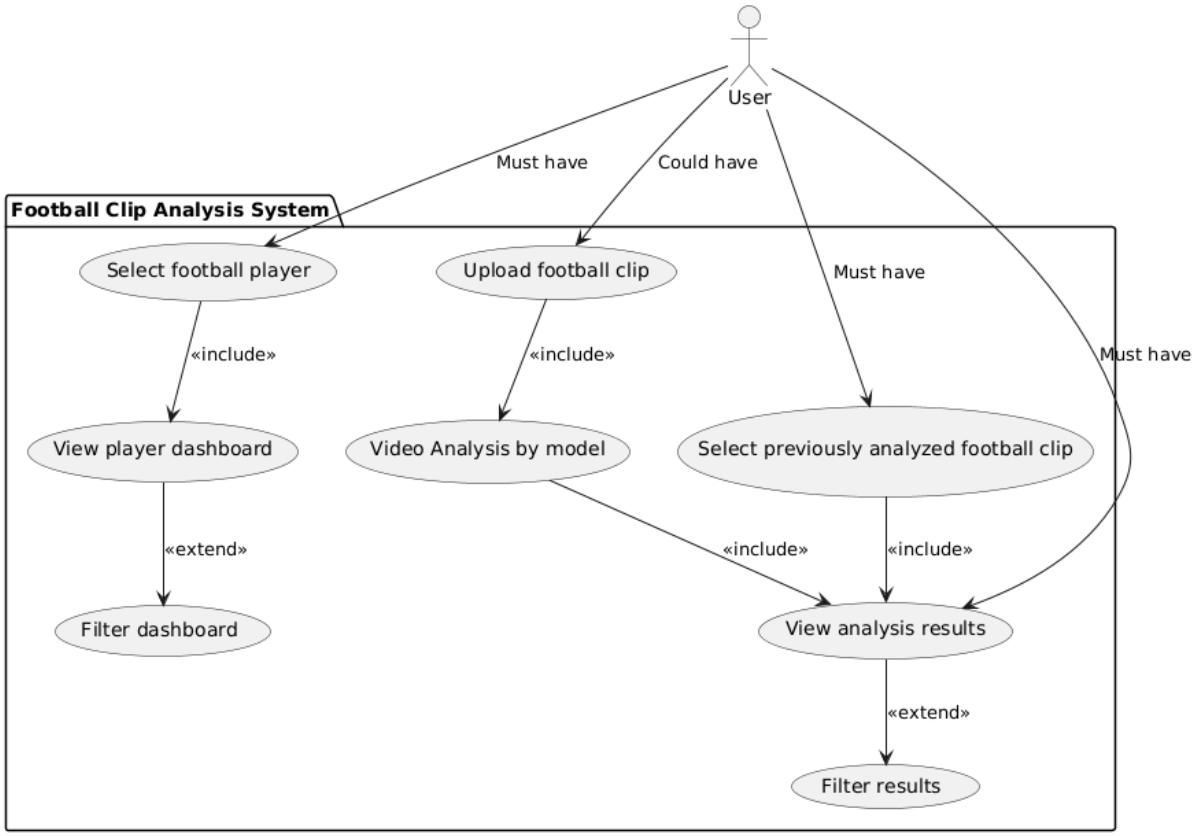

Figure 1: Project Schedule Gannt Chart

3.3 Non-functional Requirements

The following section breaks down the non-functional requirements of the project, which were grouped into URPS categories.

3.3.1 Usability

End-user interface: The graphical user interface (GUI) is designed to be intuitive and aesthetically pleasing, encompassing two primary windows: the Video Analysis Window and the Player Dashboard. In the Video Analysis Window, users have the option to select a previously analyzed video clip or upload a new clip for analysis by the model. Upon completion of the analysis, the system will present insights in a clear and organized manner, facilitating easy interpretation of the results. The Player Dashboard allows users to search for specific football players. Once a player is selected, the dashboard will display comprehensive information and statistics relevant to that player. Users can further refine the displayed data through various filtering options, enabling them to tailor the information to their specific needs.

Documentation: The documentation will be developed in parallel with the application, providing a comprehensive description of the program's entire structure. This approach will ensure that future enhancements can be implemented smoothly and efficiently, supporting ongoing development and scalability.

3.3.2 Reliability

The application will undergo comprehensive testing to identify and eliminate any errors that may hinder user functionality. This thorough evaluation is essential to ensure a smooth and uninterrupted user experience. Additionally, the analysis model will be rigorously tested to guarantee its reliable operation, ensuring it runs without issues consistently. This dual focus on application stability and model performance is crucial for delivering accurate insights to the user.

3.3.3 Performance

The model employed for clip analysis will undergo rigorous accuracy testing to ensure that it consistently delivers high-value insights. This validation process aims to guarantee repeatability and reliability in the analysis outcomes. Furthermore, the entire application is designed to be responsive, ensuring seamless performance without any delays or errors. This focus on responsiveness is critical for enhancing user experience and facilitating efficient data analysis.

3.3.4 Supportability

The application should be designed to allow for seamless and easy scalability, ensuring that future extensions or enhancements can be implemented without issues. This includes the flexibility to adjust the analysis model and modifying the independent components of the application, such as the user interface (GUI) or dashboard. Comprehensive and well-structured documentation will support these updates, enabling smooth integration and continuous improvement.

4 Project Schedule

No	Task	Author	Start Date	End Date	Status
1	Project Topic Definition	Osman Aliyev Hubert Jaczyński Aleksander Klepka	30/09/2024	2/10/2024	✓
2	Project Details Planning	Osman Aliyev Hubert Jaczyński Aleksander Klepka	3/10/2024	10/10/2024	✓
3	Project Topic Description	Osman Aliyev Hubert Jaczyński Aleksander Klepka	11/10/2024	17/10/2024	✓
4	Problem Analysis and Research	Osman Aliyev Hubert Jaczyński Aleksander Klepka	12/10/2024	28/10/2024	
5	Project Solution Description	Osman Aliyev Hubert Jaczyński Aleksander Klepka	29/10/2024	7/11/2024	
6	Data Collection and Preparation	Osman Aliyev Hubert Jaczyński Aleksander Klepka	8/11/2024	10/11/2024	
7	Fine Tuning Existing Model	Hubert Jaczyński Aleksander Klepka	11/11/2024	13/11/2024	
8	GUI Implementation	Osman Aliyev Aleksander Klepka	14/11/2024	17/11/2024	
9	Dashboard Implementation	Osman Aliyev Aleksander Klepka	18/11/2024	20/11/2024	
10	Model Implementation	Hubert Jaczyński Aleksander Klepka	21/11/2024	5/12/2024	
11	Unit Testing	Osman Aliyev Hubert Jaczyński Aleksander Klepka	6/12/2024	9/12/2024	
12	Integration, System and Acceptance Testing	Osman Aliyev Hubert Jaczyński Aleksander Klepka	10/12/2024	14/12/2024	
13	Project Evaluation and Final Documentation	Osman Aliyev Hubert Jaczyński Aleksander Klepka	15/12/2024	20/12/2024	

Table 3: Project Schedule

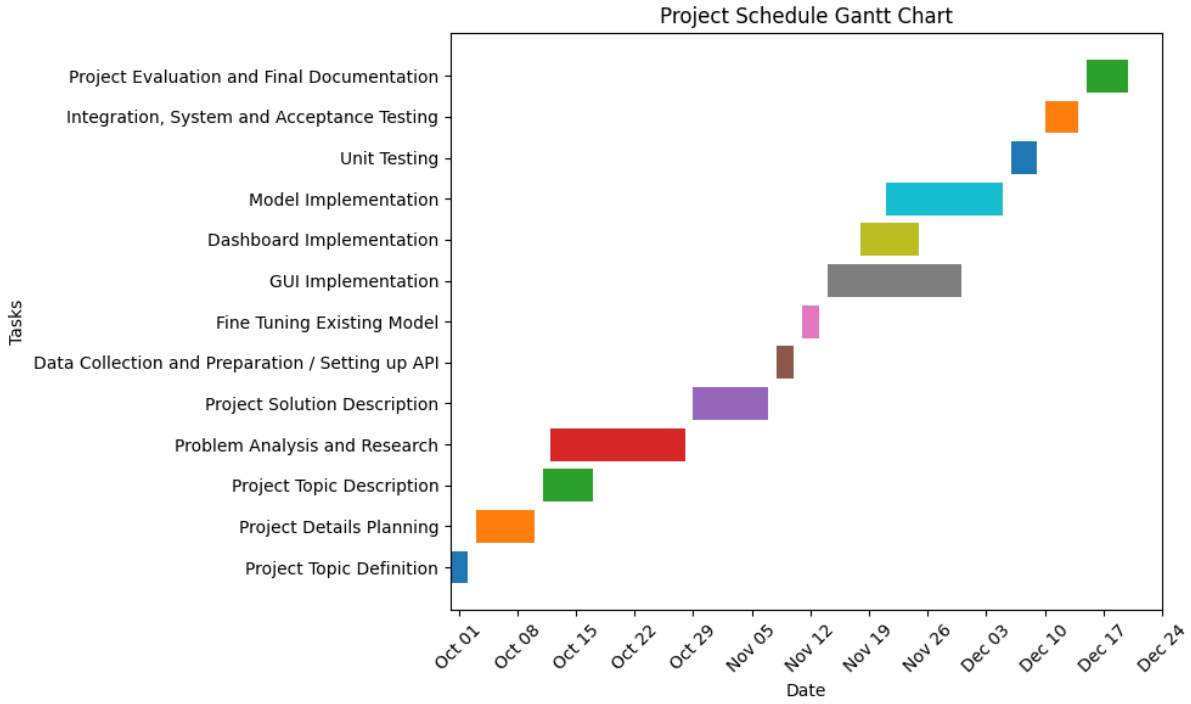

Figure 2: Project Schedule Gantt Chart

5 Risk Analysis

SWOT analysis was utilized to conduct a thorough risk assessment of the project.

5.1 Strengths (Internal)

- History of successful collaborations on past projects.
- Effective time management and leadership support which drives goals completion.
- Experience in Machine Learning, Python, Front End Development, and overall Football knowledge.

5.2 Weaknesses (Internal)

- Set and limited amount of time to complete the project.
- Delays in the predefined schedule.
- Spending a considerable amount of time adapting to new technologies.

5.3 Opportunities (External)

- Persistent and insightful support from the supervisor.
- Rich amount of highly informative and widely available resources related to the topic of the project.
- Significant value of the project, which will contirbue to professional development.

5.4 Threats (External)

- Lack of efficient computing power.
- Poor data quality.
- Incompatibility between certain components of the project.

A detailed description of weaknesses and threats can be found in the section below.

5.5 Activities to minimize the chances of weaknesses and threats

5.5.1 Set and limited amount of time to complete the project

Value: Very unlikely.

Source of Weakness: Other projects to prepare, personal commitments, and other responsibilities.

Activities to minimize chances of occurrence: Approach the project with a clear plan, set milestones, and adhere to the predefined schedule. Systematic work.

5.5.2 Spending a considerable amount of time adapting to new technologies

Value: Likely.

Source of Weakness: Working first time with some of the technologies. Trying to learn everything at once.

Activities to minimize chances of occurrence: Learning how to use some of the libraries and technologies on small projects beforehand and applying solutions that we have grasp on.

5.5.3 Lack of efficient computing power

Value: Very Likely.

Source of Weakness: Model training and fine-tuning requires a lot of computing power and memory, which might be difficult to attain on the equipment we work on.

Activities to minimize chances of occurrence: Utilizng Google Colab, or getting access to the faculty supercomputer.

5.5.4 Poor data quality

Value: Unlikely.

Source of Weakness: Provided videos might have low resolution, different camera angles.

Activities to minimize chances of occurrence: Properly choosing the videos that we attained. Normalizing the data.

5.5.5 Incompatibility between certain components of the project

Value: Possible.

Source of Weakness: Different versioning of libraries that we utilize on Google Colab and on our machines.

Activities to minimize chances of occurrence: Investigating what libraries versions are and using the same ones on our machines. Creating a virtual environment and having requirements.txt file in the repository so that each member can install the same versions of libraries.

References

[1] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi (2016). *You Only Look Once: Unified, Real-Time Object Detection*. Institue of Electrical and Electronics Engineers (IEEE).

[2] Peters, J. F. (2017). *Foundations of Computer Vision: Computational Geometry, Visual Image Structures and Object Shape Detection*. Springer International Publishing AG.

[3] DFL Bundesliga Data Shootout. (2022). Bundesliga.
Available at: `https://www.bundesliga.com/en/aws-bundesliga-data-shootout` (last accessed: October 22, 2024).

[4] Kelleher, John D. (2019). *Deep Learning*. MIT Press.

[5] Ian Goodfellow and Yoshua Bengio and Aaron Courville (2016). *Deep Learning*. MIT Press.

[6] Gary Bradski, Adrian Kaehler (2008). *Learning OpenCV*. O'Reilly Media, Inc.

[7] Mohammad Faizar (2022). *Roboflow*.
Available at: `https://medium.com/red-buffer/roboflow-d4e8c4b52515` (last accessed: October 22, 2024).

[8] David Cariboo (n.d.). *Football Data from Transfermarkt*. Available at: `https://www.kaggle.com/datasets/davidcariboo/player-scores` (last accessed: October 22, 2024).