HACETTEPE UNIVERSITY
ENGINEERING FACULTY
DEPARTMENT OF COMPUTER ENGINEERING

BBM 325
INTERNSHIP REPORT

GÜLVERA YAZILITAŞ
2210356111

Performed at

Nuun Medya ve Yazılım

01.07.2024 – 12.08.2024
30 Workdays

Report template version: v1. Feb 27, 2019.

TABLE OF CONTENTS

1 Introduction

I completed my internship at Nuun Creative (https://nuuncreative.com/) in the Software Development Department. Nuun Creative focuses on developing innovative solutions in various fields. My motivation for choosing this company stemmed from their involvement in developing an AI-based football analysis program.

For about a year, I had been particularly interested in real-life projects supported by AI, and I believed this internship would not only enhance my current skillset but also provide an opportunity to apply the theoretical knowledge I had gained during my studies. Additionally, I was eager to learn new techniques and tools by participating in a practical and impactful project.

During my internship, I contributed to the development of algorithms that generate heatmaps for football matches and calculate the ball possession percentages for each team. These features were implemented using AI technologies. This experience allowed me to practice and improve my skills while gaining valuable insight into AI applications in real-world scenarios.

2 Company Information

2.1 About the company

Nuun Creative is a digital agency headquartered in Ankara, Turkey. The company specializes in enhancing clients' digital presence, catering to both local and international markets.

Nuun Creative operates in various digital service domains:

Web Design: Creating websites tailored to clients' brand identity.

SEO Optimization: Improving website rankings in search engines.

Social Media Management: Managing and growing social media accounts.

Advertising Management: Handling digital marketing campaigns.

Graphic and Video Design: Developing visual and multimedia content.

Custom Software: Building tailored software solutions

Main Target Areas and Products

The company focuses on businesses aiming to enhance their digital visibility and developing software applications. Its products include customized websites, e-commerce platforms, social media campaigns, and promotional video productions, targeting both SMEs and large enterprises.

Key stakeholders include the company's clients across various industries, its in-house team of professionals, and strategic solution partners who collaborate on larger projects.

2.2 About your department

My department is the Software Development department.

Department organizational chart

- 1 Computer Engineer (Team Leader)
- 1 Graphic Designer
- 2 Internship Students (me and my teammate)

Your position in the team

My position included implementing required algorithms, training object detection model, and testing performance of our program.

Main topics and products

My department's main topics were developing software applications, AI integrated software solutions and meeting efficiency requirements.

2.3 About the hardware and software systems

The software department at Nuun Creative employs a comprehensive range of hardware and software tools to ensure the delivery of top-notch digital services. These resources include:

- **Web Development Tools:** Modern integrated development environments (IDEs) such as Visual Studio Code, coupled with advanced frameworks like React.js, Angular, and Node.js, enable the creation of responsive and scalable websites. The team also leverages robust servers and cloud platforms for efficient hosting and testing of their projects.
- **Design Software:** Visual content is crafted using industry-standard software like Adobe Photoshop and Illustrator, ensuring high-quality designs for graphics and advertising materials.
- **AI and Data Analysis Tools:** Proprietary AI-driven applications are used to automate tasks and enhance operational efficiency.
- **Project Management Platforms:** Collaborative tools like Trello and Asana are employed for task coordination and workflow management.
- **Social Media Tools:** Advanced software supports scheduling posts, analyzing performance metrics, and optimizing content for search engines, ensuring successful social media campaigns.

These tools and systems form the backbone of the department's ability to deliver innovative and efficient solutions to their clients.

2.4 About your supervisor

- **Name**: Civan Ünver
- **Address**: Ostim Mh. 1308 Cd. No 9/17 Daire No:4 Yenimahalle/Ankara
- **Telephone Number:** 0531 457 97 16
- **Email Address**: civan.unver@gmail.com
- **Education**: Kırıkkale Üniversitesi Bilgisayar Mühendisliği / 2022 Graduate

3 Work Done

Our project involved developing an AI-driven football match analysis program. We were tasked with obtaining football match videos from a YouTube channel and applying the AI algorithms we implemented to process them. For each game, we generated two analysis charts—one for each half. These charts included a heatmap of the field, highlighting the areas with the most player activity, and percentages indicating each team's ball possession and interactions. The program should detect all the players by object detection and find their location on the field for the heatmap part. It should distinguish the players by looking at the color of their uniforms and detect the ball during the game for the second part. My teammate and I completed the whole project together. The motivation of this project was to provide detailed and reliable statistics to the league teams so that they could improve their game strategies according to the outcome of our AI analyses program. Various programs like this are widely used in football clubs around the world. We started our project by training a machine learning model to detect players and the ball in football match videos. We wrote our code in Python and used Visual Studio Code as our IDE.

Choosing an Object Detection Algorithm

After researching different object detection algorithms, we decided to use YOLOv8. We chose YOLOv8 because it provides a good balance between speed and accuracy, which is important for real-time object detection.

Preparing the Data

To prepare the dataset for training our object detection model, we began by extracting individual frames from the match videos using **OpenCV**. OpenCV's video processing capabilities allowed us to efficiently split the videos into thousands of frames, ensuring we had a diverse set of data capturing various moments of the matches.

Each frame was then manually annotated using a labeling tool called **LabelImg**, where we marked the positions of objects such as players and the ball. This meticulous labeling process helped create an accurate dataset with detailed bounding boxes for the objects of interest.

Once the labeling was complete, we utilized **Roboflow** to organize and preprocess the data. Roboflow facilitated splitting the dataset into **training, validation, and test sets**, while also applying essential augmentations such as resizing, rotation, and flipping to enhance model generalization.
In summary, the workflow combined **OpenCV** for frame extraction, **LabelImg** for annotation, and **Roboflow** for dataset management and preprocessing, ensuring a high-quality dataset ready for model training.

Training the Model

To train our object detection model, we utilized **Google Colab**, a cloud-based platform offering access to high-performance GPUs and TPUs. This allowed us to train our model efficiently without the need for expensive local hardware. The training process involved feeding the preprocessed datasets (created earlier using Roboflow) into the model. We implemented techniques like learning rate adjustment, data augmentation, and early stopping to enhance the model's performance and prevent overfitting.
After completing the training, we rigorously tested the model to evaluate its ability to accurately detect players and the ball in the videos. Testing was conducted on unseen frames to ensure the model's robustness in identifying objects under varying conditions, such as different camera angles, lighting, and player movements. Based on these tests, we iteratively fine-tuned the model parameters to achieve optimal detection accuracy.

Applying Homography Matrix

The next step in our project involved the application of a **homography matrix** to transform the detected positions of players and the ball into a bird's-eye view representation of the football field. This transformation required a solid understanding of homography concepts, as it involves mapping points from one perspective (camera view) to another (field view).
To calculate the homography matrix, we identified **at least four reference points** on the football field that are clearly visible in both the camera's perspective and the actual field layout. These reference points included the **field corners**, the **midfield point**, **goalposts**, and the **penalty area corners**. Using these points, we applied **OpenCV's findHomography() function** to compute the transformation matrix.
By applying this matrix to the detected positions of players and the ball, we were able to project their coordinates onto a scaled bird's-eye view of the field. This step was crucial for subsequent tasks like analyzing player movement patterns, generating heatmaps, and performing advanced spatial analyses. The accuracy of the transformation was validated by cross-referencing known field dimensions and testing the alignment of projected positions with the actual field layout.

Visualizing the Results

We then visualized the positions of the players on a bird's-eye view of the field using the transformed coordinates. This helped us to better analyze and understand player positions in a match.

It took us three weeks to complete our algorithm that determines player positions in a bird's-eye view of the field and generates a heatmap for every match. The final result provides useful insights into player positioning and movements during the game.

In the second part of the project, we focused on calculating the percentages of each team's interaction with the ball. To achieve this, we needed to identify the uniform colors of the players to determine which team member was closest to the ball at any given time.

Identifying Player Uniform Colors

Using the bounding box coordinates detected by the YOLO model, we extracted the regions surrounding each player. From these regions, we focused on the **upper-central part of each bounding box**, as it typically represents the player's jersey color, which is a distinguishing feature.

To determine the dominant color in this area, we leveraged **OpenCV** to process the pixel data. Specifically, we used color histograms and converted the image to the HSV color space, which simplifies color-based analysis by separating chromatic content (hue) from intensity (value). This made it easier to identify and isolate the most prominent color in each region.

After extracting the dominant color, we compared it against a predefined set of team colors. These team colors were preloaded into the system as reference points. By measuring the **Euclidean distance** in the HSV color space, we were able to match each player’s uniform with the corresponding team color, effectively assigning them to their respective teams.

Clustering Colors with the K-Means Algorithm

To improve the accuracy of team classification, we applied the **K-Means clustering algorithm**. This algorithm grouped detected uniform colors into two clusters, representing the two teams on the field. Here's how the process worked:

1. **Feature Extraction**: The pixel data of the dominant color from each bounding box was used as input features for the clustering algorithm.
2. **Clustering Process**: K-Means grouped these color features into two clusters, minimizing the variance within each group.
3. **Cluster Assignment**: Each player was assigned to a cluster, with the clusters aligned to the known team colors based on centroids calculated during the clustering process.

This clustering step enhanced uniform classification, even in cases where lighting conditions or camera angles caused slight variations in the detected colors.

Calculating Distances to the Ball

After correctly identifying and clustering the players by team, we calculated the distances of each player to the ball for each frame. By analyzing these distances over time, we could determine which team was closest to the ball most frequently. This enabled us to calculate the percentage of ball possession for each team.

Increasing the Accuracy of Our Models

For get higher accuracy from exact object detection we have trained four different YOLO models. Each model focuses on different features: Model for detecting players, model for detecting the ball, model for detecting corners of (left and right) goals and, central and upper point of the midfield point, and a model for detecting corners of penalty areas. By combining these models and tuning their training parameters we have achieved accuracy we wanted. And at some point, we needed more data than we have so we have applied Roboflow data augmentation. It has changed our results significantly.

Usage of Our Program

We arrange our code to make it cleaner and more readable. And models that we trained are provided in a folder with our *heatmap.ipynb* file. All they need to do is give the match video path and wait a while. Then our heatmap results of the match are ready to be analyzed.

Individual Work

Throughout this project, I have made significant contributions, particularly in the development of algorithms, model training, and data labeling. I was responsible for labeling thousands of images, which formed the foundation for training and validating the models. This involved accurately annotating football field references, players, and the ball, which required a meticulous approach to ensure the models were trained with high-quality data.

I developed various algorithms, focusing on object detection and tracking, including the use of YOLO for player and ball detection. I also fine-tuned and trained the models, optimizing them for better accuracy and performance. In addition to model training, I worked on implementing homography transformations to map the detected player positions onto the bird's-eye view of the field.

Moreover, I handled the creation of heat maps by analyzing player movements across thousands of frames, ensuring that the generated visualizations provided meaningful insights into player distribution and activity.

Through this process, I gained hands-on experience in data processing, model development, and evaluation, while significantly contributing to the overall progress of the project.

4 Performance

+d Outcomes

4.1 Applying Knowledge and Skills Learned at Hacettepe

The knowledge and skills I gained at Hacettepe University significantly impacted my performance during my internship. In my first year, through the BBM101 course, I was introduced to my first machine learning algorithm using Python. This experience provided me with insights into machine learning models and their workings. In my second year, I participated in a Teknofest project as part of a student community at Hacettepe. The project focused on AI applications in transportation, marking my first exposure to object detection algorithms and models such as YOLO. During this project, I learned how to preprocess data for training machine learning models and understood the influence of various parameters on model performance. These experiences proved invaluable during my internship. The skills and knowledge I had acquired enabled me to contribute effectively to the project. Moreover, the internship allowed me to enhance my existing skills while also gaining new expertise in AI and its practical applications.

4.2 Solving Engineering Problems

Which engineering problems did you solve related to the computer systems and applications during your internship? Explain in detail.
During the project, I encountered several engineering challenges related to computer systems and applications. Below are the key problems I faced and how I addressed them:

1. Problem: Accurate Homography Estimation
When transforming player positions from the video frames to a bird's-eye view, achieving accurate homography was challenging, especially due to variations in camera angles and incomplete field visibility.
Solution:

- Used a YOLO model to detect specific reference points on the football field (e.g., penalty box corners, center circle).
- Improved accuracy by training the model on diverse datasets to handle variations in lighting, angles, and occlusions.
- Verified the results by overlaying transformed coordinates on a base field image and iteratively adjusted the homography matrix for alignment.

2. Problem: Uniform Color Identification
Identifying player uniforms was difficult due to lighting changes, overlapping players, and motion blur.
Solution:

- Focused on the upper-central region of bounding boxes for color extraction, avoiding areas with potential noise like legs or backgrounds.
- Converted images to HSV color space to separate chromatic content from intensity for better color detection.
- Applied K-Means clustering to group colors into team categories, reducing classification errors under varying conditions.

3. Problem: Processing High Volumes of Data Efficiently
Analyzing thousands of frames and calculating metrics (e.g., ball possession, heatmaps) required significant computational resources.
Solution:

- Implemented batch processing for frame-by-frame data analysis, reducing memory usage.
- Used optimized libraries like OpenCV for image processing and NumPy for numerical computations.
- Parallelized operations where possible to leverage multi-core CPUs, speeding up the overall pipeline.

4. Problem: Heatmap and Possession Calculation Accuracy
Generating accurate heatmaps and ball possession percentages required integrating spatial and temporal data without introducing errors.
Solution:

- Normalized all player positions using the same homography matrix to ensure alignment with the base field image.
- Validated heatmap outputs by comparing them against expected activity zones based on match footage.
- Used Euclidean distance calculations for possession metrics and verified results through manual sampling.

4.3 Teamwork

Our team had 4 members during my internship: a computer engineer (team leader), a graphic designer, and two interns. The graphic designer's role was to determine the design of the outcome of our program. She focused on making our result charts more user-friendly and readable. Our team leader managed the entire process and assigned tasks to everyone. My intern colleague and I were asked to implement the program they wanted. Throughout the entire internship, we worked together in a peaceful environment. We conducted the necessary research,

and when a problem arose, we consulted our team leader. He provided us with the required resources.
As interns, my colleague and I were second-year students tasked with implementing the core functionality of the program. This included coding, debugging, and integrating the required features. Despite our limited experience, we were entrusted with critical responsibilities, which motivated us to perform at our best.
The team dynamic was very positive, fostering open communication and collaboration. We worked together in a supportive and peaceful environment, ensuring that everyone could share their ideas and resolve problems collectively. When we encountered challenges or technical difficulties, we sought advice from our team leader, who always provided helpful solutions and guidance.
Overall, this teamwork experience allowed me to grow both technically and personally, as we successfully completed the project while maintaining a harmonious and productive work atmosphere.

4.4 Multi-Disciplinary Work

I was not a part of a multi-disciplinary work. The team consisted of computer engineers and computer engineering students.

4.5 Professional and Ethical Issues

During my internship, I had to work remotely for a few days due to family issues. However, our team leader (who was also my supervisor) provided me with an online work environment where I could collaborate with my teammates. This situation was handled professionally, and the workload was divided equally among the team, so the project was not affected.

4.6 Impact of Engineering Solutions

During the internship, I learned about the various impacts of engineering solutions, particularly in the context of the AI-based football analysis project. Economically, the use of AI and machine learning to analyze football matches can potentially revolutionize sports analytics, offering clubs and broadcasters more precise insights into player performance, match strategies, and game predictions. This can lead to better decision-making, which could reduce costs and increase efficiency in scouting and team management.
Environmentally, while AI solutions generally require considerable computing resources, the use of cloud computing services in a remote working environment helped mitigate the environmental impact by reducing the need for local data storage and on-site infrastructure.

4.7 Locating Sources and Self-Learning

Explain the self-learning that you did during your internship. You should mention any sources that you located and how you found them (this would include Web sites, books, journals, experts, etc), and what part of your internship task you needed them for.

During my internship, I engaged in self-learning to enhance my skills and effectively contribute to the project. I relied on several sources to expand my knowledge in areas like computer vision and AI in sports analysis. Websites like Stack Overflow, Towards Data Science, and Medium provided helpful tutorials and articles on using OpenCV, YOLO, and machine learning techniques. For understanding the theoretical background, I read academic journals and research papers on AI applications in sports. I also referred to documentation for OpenCV and Keras when implementing machine learning models. I needed these resources primarily for coding, debugging, and improving the accuracy of the models. The most valuable sources were online forums and tutorials, which allowed me to resolve specific technical challenges related to image processing and model training.

4.8 Using New Tools and Technologies

During my internship, I utilized several new tools and technologies to develop and implement the football analysis program. These included LabelImg, Roboflow, OpenCV, YOLOv8, TensorFlow, and Google Colab.

1. **LabelImg**: I used this tool for annotating the images of players and the ball in video frames. This tool helped in creating the labeled dataset necessary for training the YOLOv8 model.
2. **Roboflow**: I used Roboflow to preprocess and augment the images, which enhanced the model's ability to recognize players and the ball from different angles and lighting conditions.
3. **OpenCV**: OpenCV was essential for image processing tasks, such as extracting color features from the player's uniform and processing frames for homography transformation.
4. **YOLOv8**: I implemented YOLOv8, a state-of-the-art object detection model, to detect players and the ball in video frames in real-time, enabling accurate tracking and positioning analysis.
5. **Google Colab**: Google Colab provided the cloud-based environment necessary for training the models and performing resource-intensive computations without the need for powerful local hardware.

By the end of the internship, I became proficient in using these tools and integrated them effectively to accomplish the project goals.

5 Conclusions

During my internship, I contributed to the development of an AI-based football analysis project. My primary role involved implementing algorithms to generate heatmaps that visualize player activity across the football field and calculating ball interaction percentages for each team. I utilized YOLO-based object detection models for identifying players and the ball and applied homography transformations to map detections onto a bird's-eye view of the field.

References

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. 2016. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 779-788: https://doi.org/10.1109/CVPR.2016.91

[2] Romero-González, J., and Monroy, A. 2023. A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. Mach. Learn. Knowl. Extr. 5, 4 (Nov. 2023), 1680-1716 : https://doi.org/10.3390/make5040083

[3] R. Fan et al., "Learning Collision-Free Space Detection From Stereo Images: Homography Matrix Brings Better Data Augmentation," in IEEE/ASME Transactions on Mechatronics, vol. 27, no. 1, pp. 225-233, Feb. 2022, doi: 10.1109/TMECH.2021.3061077.

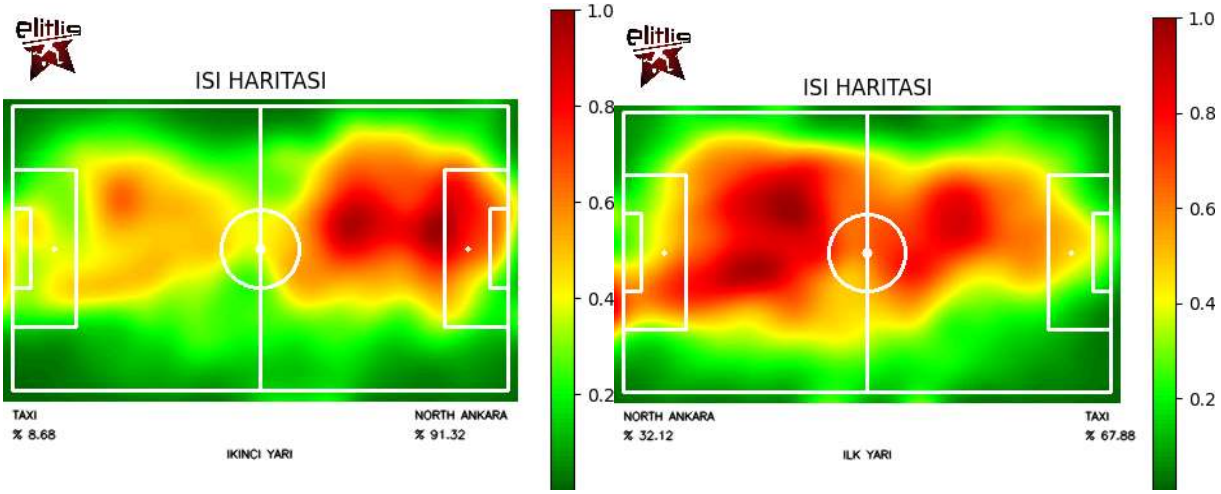
[4] N. D. Sugiharto, D. Elbert, J. Arnold, I. S. Edbert and D. Suhartono, "Mall Customer Clustering Using Gaussian Mixture Model, K-Means, and BIRCH Algorithm," 2023 6th International Conference on Information and Communications Technology (ICOIACT), Yogyakarta, Indonesia, 2023, pp. 212-217, doi: 10.1109/ICOIACT59844.2023.10455950.

[5] Y. Wang, T. Cai and O. Zatarain, "Objects Detection and Recognition in Videos for Sequence Learning," 2019 IEEE 18th International Conference on Cognitive Informatics & Cognitive Computing (ICCI*CC), Milan, Italy, 2019, pp. 211-216, doi: 10.1109/ICCICC46617.2019.9146073.

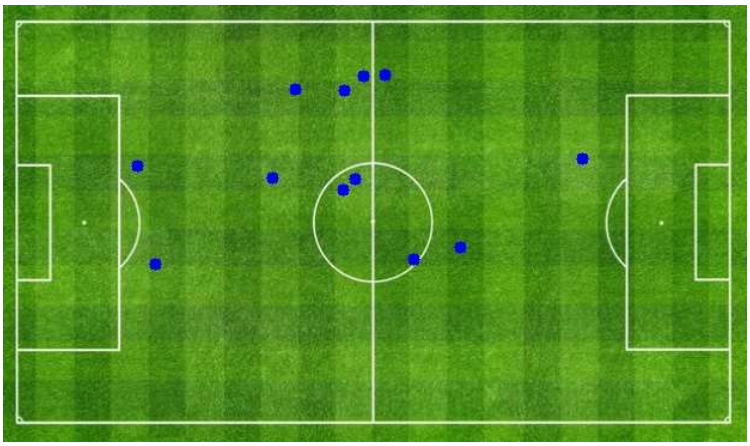
[6] Abdullah Tarek, "Football Analysis," GitHub, 2024. Available: https://github.com/abdullahtarek/football_analysis.

Appendices

These are two heatmap of a match. These were generated with our program. It takes match video from Youtube and calculate heatmap and ball possession of each team.
The Match video: https://www.youtube.com/watch?v=-ychJNGqXjE



After applying homography matrix to our video frames we got this for each frame



Here are a few screenshots from one of our output videos. These visuals demonstrate the results of our model testing. The model successfully detected players and the ball on the field and identified each player's team based on the color of their uniforms.