Football Players Tracking from Multiple Cameras using Deep Learning

Kerollos Wageeh Youssef Saleeb 193468
Youssef George Fouad Saad 19P9824
Anthony Amgad Fayek Selim 19P9880
Marc Nagy Nasry Sorial 19P3041
Youssef Ashraf Mounir Showeter 19P4179

Supervisor: **Prof. Dr. Mahmoud I. Khalil**
Sponsored by: **KoraStats**

Ain Shams University, Faculty of Engineering ICHEP
Computer Engineering and Software Systems

January 2024

Abstract

In this comprehensive thesis, we delve into the intricate development of a sophisticated deep learning and computer vision system tailored for the dynamic realm of football player tracking during matches. The thesis meticulously explores various image processing and tracking methodologies, placing particular emphasis on crucial aspects such as player re-identification, occlusion resolution, and similarity matching.

The core outcome of our system is the extraction of high-fidelity spatio-temporal trajectories of players, thereby unlocking a myriad of possibilities for statistical analysis. These trajectories serve as the foundation for generating in-depth player statistics, offering a wealth of valuable insights for sports enthusiasts and analysts alike. The envisioned system not only automates the tracking process but also paves the way for a transformative user experience through a dynamic and graphically rich interface. While our system excels in autonomous tracking, we acknowledge the existence of critical scenarios that may require human intervention to rectify miss-trackings. Thus, human supervision remains an integral part of the process, highlighting the collaborative nature of our approach.

The thesis unfolds with a detailed discussion on the intricacies of image processing, shedding light on the methods employed. Our systematic approach ensures a robust and accurate player tracking system, capable of adapting to the fast-paced and dynamic nature of football matches.

The experimental phase, constituting the first half of our team's graduation project, serves as a crucial preliminary stage. This phase lays the groundwork for a full-fledged working system, culminating in a dynamic user interface that will stand as the crowning achievement of this ambitious project. Through this thesis, we aim to contribute significantly to the intersection of technology and football analytics, revolutionizing player tracking and analysis. Our journey unfolds as a testament to the fusion of technological innovation and sports expertise, promising a transformative impact on the landscape of football analytics.

Keywords: Football Analytics, Player Tracking, Computer Vision, Deep Learning.

Acknowledgments

We would like to express our sincere gratitude to all those who have contributed to the completion of the first half of this project and thesis.

First and foremost, we extend our appreciation to our supervisor, Prof. Dr. Mahmoud I. Khalil, for his invaluable guidance, support, and encouragement throughout the entire project. His expertise and insights have significantly enriched our understanding and enhanced the quality of this work.

We are grateful to **KoraStats** for their sponsorship and collaboration during this project. Special thanks to Ahmed Bahgat, the Research and Development Officer, for his technical support and insights that hugely contributed to the success of our work. Additionally, we appreciate the support of Nasr Ali, the HR Manager, for facilitating the collaboration and providing organizational support.

We express our deepest appreciation to the faculty and staff at Ain Shams University, especially the Computer Engineering and Software Systems department, for providing an environment conducive to learning and research.

Finally, we want to thank our families and friends for their unwavering support and encouragement throughout this academic journey. Thank you to everyone who has been a part of this endeavor.

List of Abbreviations

Abbreviation	Meaning
AI	Artificial Intelligence
NN	Neural Network
CNN	Convolutional Neural Network
CV	Computer Vision
DL	Deep Learning
FCN	Fully Convolutional Network
VAE	Variational Auto Encoder
RNN	Recurrent Neural Network
LSTM	Long Short Term Memory
ROI	Region of Interest
YOLO	You Only Look Once
SSN	Semantic Segmentation Network
SDS	Superpixel-based Discriminative Segmentation
MNC	Multi-task Network Cascades
ReLU	Rectified Linear Unit
KB	Knowledge Base
bbox	Bounding Box

Contents

List of Figures

Chapter 1

Introduction

In the world of football, where the passion of fans converges with the precision of technology, our motivation to embark on this thesis is driven by a desire to revolutionize player tracking and analysis. As a team of five undergraduate engineers specializing in Computer Engineering with a concentration in Data Science, we stand at the crossroads of technology and football enthusiasm.

1.1 Motivation

Football, often hailed as the beautiful game, captivates millions with its dynamic play and strategic intricacies. Yet, the manual methods employed in tracking and analyzing player movements within the sport are labor-intensive, error-prone, and fail to capture the essence of the game comprehensively.

Our motivation arises from the convergence of two passions: our love for football and our commitment to pushing the boundaries of technological innovation. As fervent football enthusiasts, we recognize the immense potential for technology to enhance the understanding and analysis of the sport we hold dear.

The shortcomings of existing manual tracking methods fuel our drive to develop a sophisticated deep learning and computer vision system. By seamlessly integrating deep learning advancements with other computer vision techniques, our goal is to provide accurate positional data for every player, at every moment, across multiple camera feeds. This system aims not only to automate and enhance player tracking but also to redefine how we perceive and analyze the beautiful game.

Guided by our tech-savvy and football-loving ethos, our thesis embarks on a journey where deep learning advancements take center stage. We envision a system comprised of five main subsystems, each harnessing the power of deep learning models seamlessly integrated with other computer vision techniques. These subsystems represent the convergence of our technical prowess and football acumen, aiming to redefine player tracking and analysis in football matches.

In essence, our motivation is rooted in the belief that the marriage of technology and football can usher in a new era of sports analytics. Through our dedication to this project, we seek to contribute to the evolution of player tracking, enrich the understanding of in-game dynamics, and provide a toolset that empowers coaches, analysts, and fans alike.

1.2 Problem Statement

In the realm of football analytics, the traditional methods of manual player tracking and analysis present several challenges that hinder the depth and accuracy of performance assessment. These challenges include:

1. **Labor-Intensive Processes:** Current practices rely heavily on manual efforts for tracking player movements and analyzing game dynamics. This approach is time-consuming, requires significant human resources, and is prone to errors.

2. **Limited Granularity:** Manual tracking often provides limited granularity in capturing the intricate details of player movements, especially in fast-paced and dynamic situations during a football match.

3. **Occlusion Issues:** Occlusions, where players obstruct each other's view, pose a substantial challenge to accurate tracking. Existing methods struggle to handle such scenarios effectively.

These challenges underscore the need for an automated and technologically advanced solution that can overcome the limitations of manual tracking. As we delve into the development of our project, which comprises distinct subsystems integrating deep learning and computer vision techniques, we aim to address these issues comprehensively. Our objective is to provide a more efficient and accurate approach to player tracking and analysis in football matches, paving the way for a new era of football analytics.

Our project aims to tackle these challenges head-on by developing a computer vision system that seamlessly tracks players across multiple camera feeds, resolves occlusions, and delivers accurate positional data. By doing so, we aspire to usher in a new era of football analytics that is not only more precise but also more accessible and less reliant on manual labor.

1.3 Outline

This thesis is structured into the following chapters, each addressing specific aspects of the research and development process:

1. **Chapter Two: Literature Review**

In this chapter, we delve into existing literature and research relevant to football analytics, player tracking, deep learning, and computer vision. We explore the advancements, methodologies, and challenges documented in the academic landscape.

2. **Chapter Three: NN Architecture Review**

   Building upon the literature review, we provide an in-depth analysis of neural network architectures pertinent to our project. This chapter lays the groundwork for the deep learning advancements that will be integrated into our deep learning computer vision system.

3. **Chapter Four: Solution Architecture**

   Here, we detail the architecture of our proposed computer vision system. Each subsystem, powered by deep learning models and computer vision techniques, is introduced and explained. This chapter serves as a blueprint for the technical implementation.

4. **Chapter Five: Software Requirements Specifications (SRS)**

   We outline the software requirements for the development of our system. This includes a comprehensive overview of the functional and non-functional requirements, as well as detailed use cases.

5. **Chapter Six: Prototype Results**

   The results obtained from the prototype implementation are presented and analyzed in this chapter. We evaluate the system's performance against predefined metrics and assess its effectiveness in real-world football match scenarios.

6. **Chapter Seven: Conclusion**

   Finally, we draw conclusions from our research and development efforts. This chapter summarizes the key findings, discusses the implications of the results, and suggests directions for future work.

By following this structured outline, we aim to provide a comprehensive exploration of our project, from the theoretical foundations to the practical implementation and evaluation.

Chapter 2

Literature Review

In this chapter, we delve into the existing literature and research pertinent to the five subsystems of our project: object detection, visual tracking, instance segmentation, similarity matching, and path prediction. The review encompasses both foundational works and recent advancements in computer vision, deep learning, and sports analytics.

2.1 Object Detection in Football Match Analysis

Object detection plays a pivotal role in understanding and analyzing dynamic scenarios, and in the context of a football match, the combination of RetinaNet and Focal Loss, which was inspired by papers like "Focal Loss for Dense Object Detection" by Tsung-Yi Lin[19], proves to be particularly effective. The exploration dives into how these techniques, implemented for football match analysis, contribute to the accurate identification and tracking of the players and the ball. Notable works also include the R-CNN family of methods, such as Faster R-CNN [24], which introduced region proposal networks to improve detection speed. YOLO (You Only Look Once) [22] is another influential approach known for its real-time object detection capabilities. Research in sports analytics has explored object detection in various sports contexts, but few focus specifically on football. A notable exception is the work by Lea et al. [17], which proposes a temporal model for player detection in soccer.

2.1.1 RetinaNet: Capturing Multi-Scale Features on the Football Field

RetinaNet, with its single-stage architecture[19] and Feature Pyramid Network (FPN), offers a robust solution for detecting objects at different scales. In the context of a football match, where players and the ball vary in size and move dynamically across the field, RetinaNet's ability to capture multi-scale features becomes invaluable. The anchor mechanism helps ensure accurate localization, even in situations where players may be partially occluded, or the ball is in motion.

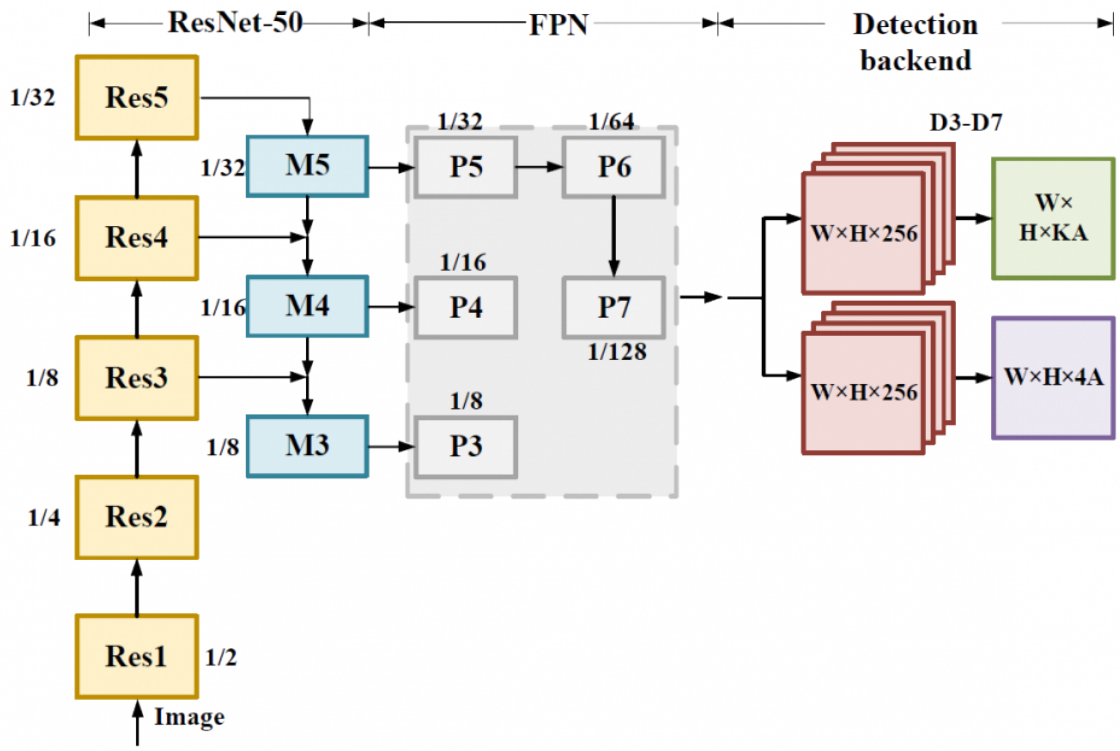


Figure 2.1: RetinaNet Architecture

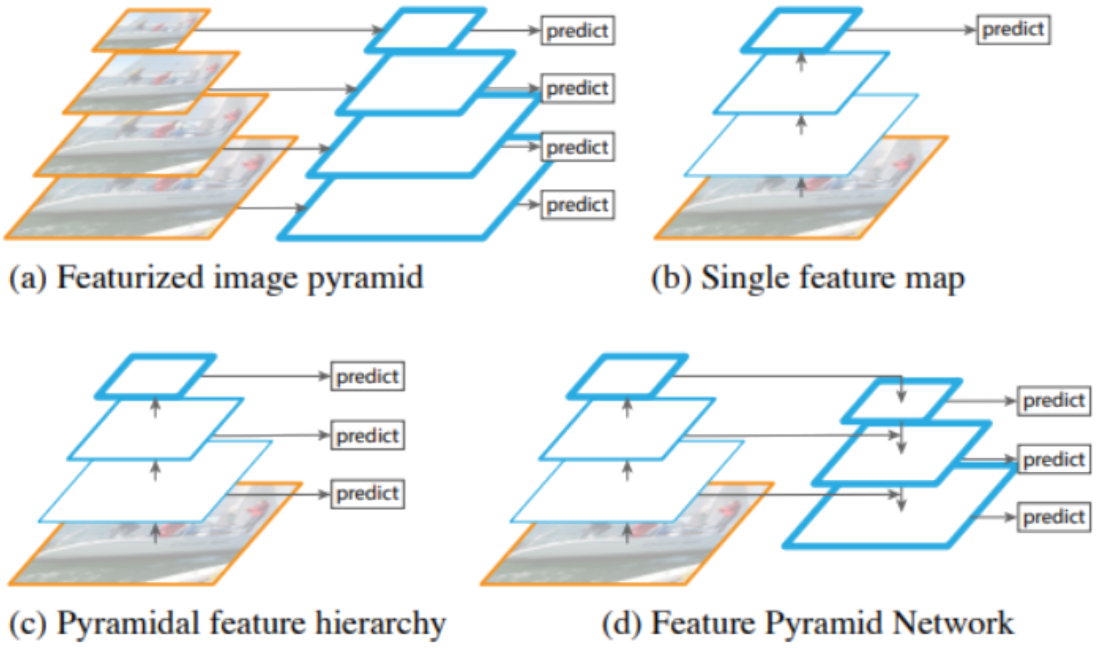


Figure 2.2: RetinaNet Steps

2.1.2 Focal Loss for Class Imbalance

The imbalanced distribution of classes in football match footage, characterized by a multitude of frames containing background information compared to instances of player or ball presence,

poses a significant challenge. Focal Loss, introduced in conjunction with RetinaNet, proves to be instrumental in addressing this class imbalance. By assigning higher weights to challenging instances, such as the ball or players in crowded regions, Focal Loss enhances the model's ability to concentrate on critical elements, thereby improving the accuracy of object detection [19].

2.2 Evolution of Instance Segmentation Approaches

Instance segmentation, a challenging task in computer vision, , involves not only detecting objects but also delineating individual instances within those objects. In particular, it enables our system to distinguish between overlapping players, enhancing the accuracy of player tracking. The Mask R-CNN architecture [10] has been influential in this domain, allowing for simultaneous object detection and segmentation. In addition, FCN-based methods, Proposal-based methods, and advancements leading up to YOLACT are indeed worth discussing when mentioning instance segmentation.

2.2.1 FCN-Based Methods

Fully Convolutional Network (FCN)-based approaches represent an early stride in instance segmentation. These methods leverage the power of convolutional neural networks to predict pixel-wise object masks [18]. In the pursuit of segmenting instances, Semantic Segmentation Networks (SSNs) such as U-Net and DeepLab demonstrated the ability to produce high-resolution object masks [3]. However, FCN-based methods faced challenges in accurately handling object boundaries and instances with varying scales, as further investigated in A.kirillov et al. "InstanceCut: from Edges to Instances with Multicut"[16].

2.2.2 Proposal-Based Methods: SDS and MNC

To address the limitations of FCN-based methods, Proposal-based methods gained prominence. Superpixel-based Discriminative Segmentation (SDS) [9] and Multi-task Network Cascades (MNC) [7] introduced a two-step approach involving object proposal generation and subsequent segmentation. SDS focused on generating high-quality object proposals, while MNC simultaneously performed object detection and segmentation tasks. Although these methods improved accuracy, they were computationally demanding.

2.2.3 Evolution to Mask R-CNN

The introduction of Mask R-CNN marked a pivotal moment in instance segmentation. Building upon the success of Faster R-CNN for object detection, Mask R-CNN extended the framework to simultaneously predict bounding boxes, class labels, and pixel-wise masks. This unified approach offered a compelling solution to the challenges of previous methods, providing accurate instance segmentation with improved efficiency [10].

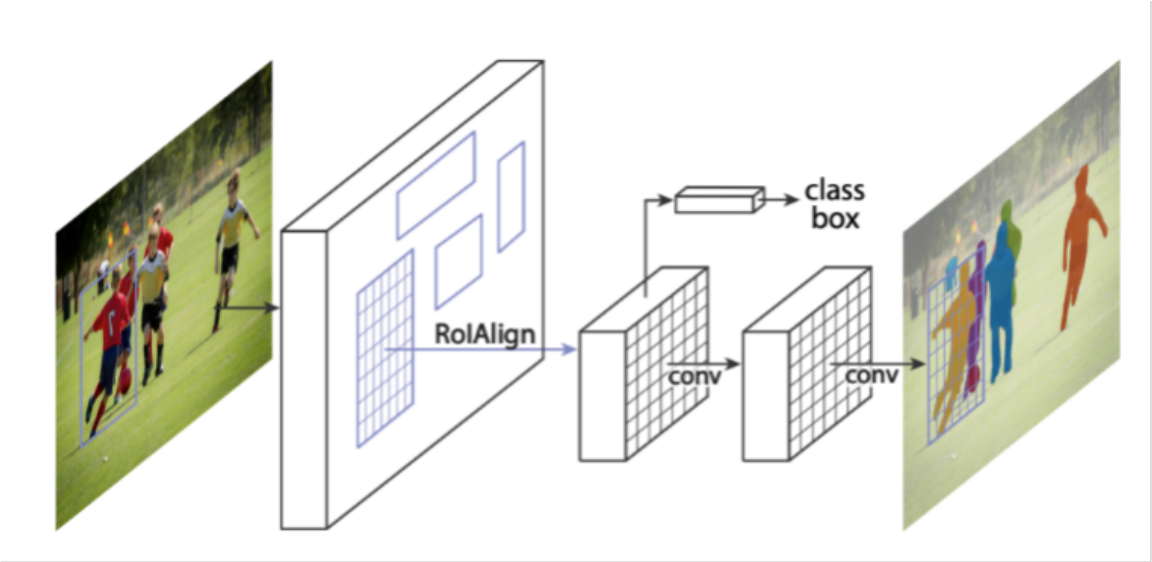


Figure 2.3: Mask R-CNN

2.2.4 Mask Scoring R-CNN

Recognizing the importance of refining mask quality, Mask Scoring R-CNN extended the Mask R-CNN architecture by introducing a mask scoring mechanism. This mechanism assessed the quality of predicted masks, allowing the model to assign higher confidence to more accurate instances. The incorporation of a mask scoring strategy enhanced the overall precision of instance segmentation results [5].

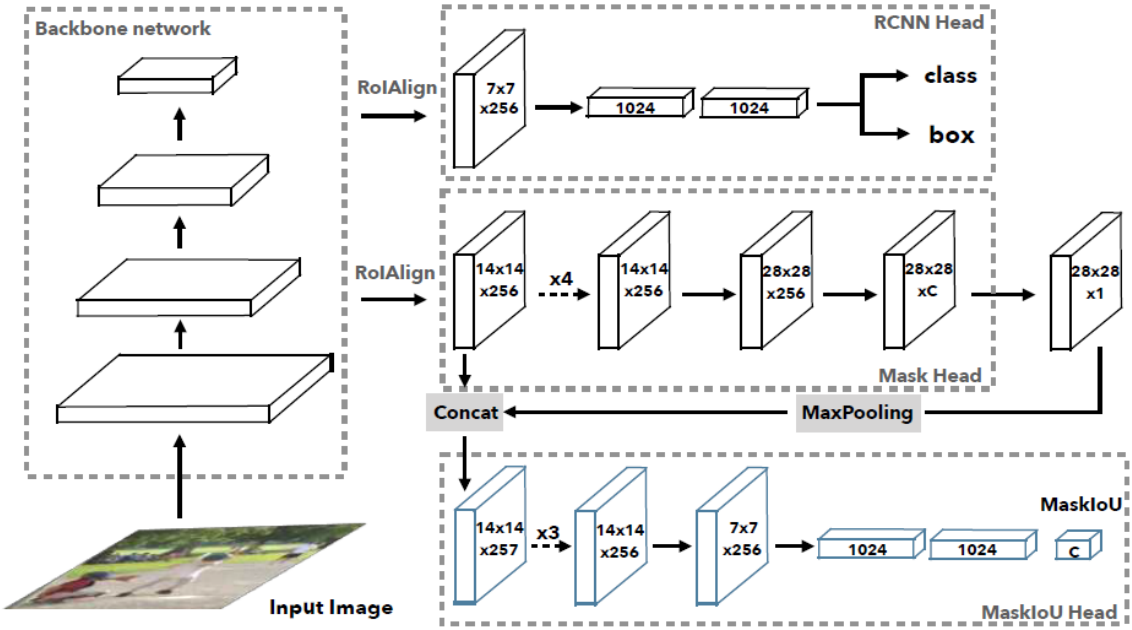


Figure 2.4: Mask Scoring R-CNN

2.2.5 YOLACT: Real-time Instance Segmentation

YOLACT represents a breakthrough in real-time instance segmentation. Departing from the traditional two-step approach, YOLACT adopts a single-shot framework, simultaneously predicting bounding boxes, class labels, and segmentation masks. This approach significantly reduces inference time while maintaining competitive accuracy. YOLACT's innovation lies in the use of prototype masks, enabling efficient computation and real-time performance [23].

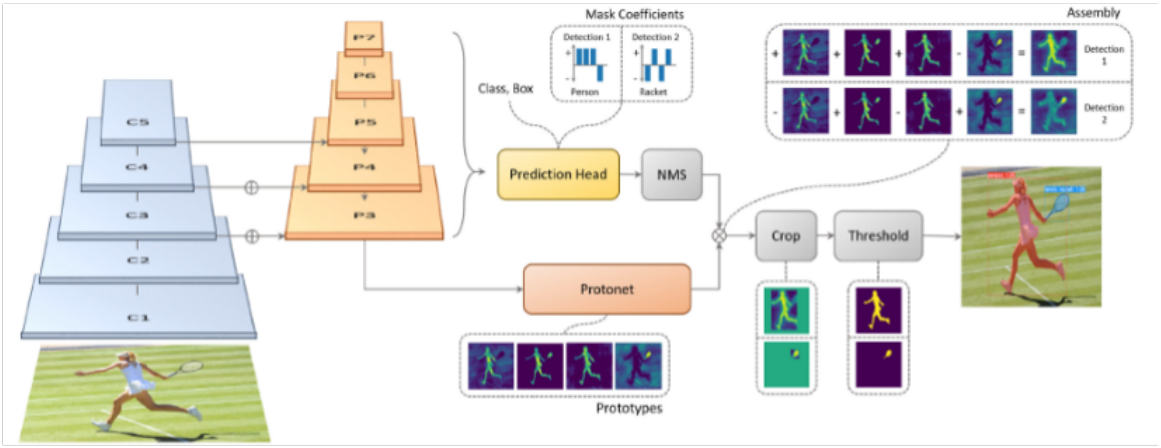


Figure 2.5: YOLACT for Instance Segmentation

2.2.6 A Comparative Analysis

A comparative analysis of these instance segmentation approaches reveals a progression from FCN-based methods to more sophisticated frameworks like YOLACT. While FCN-based methods laid the foundation for pixel-wise predictions, Proposal-based methods introduced a crucial refinement step. The evolution continued with Mask R-CNN, emphasizing unified prediction, and advanced further with Mask Scoring R-CNN and YOLACT, addressing nuances in mask quality and real-time performance.

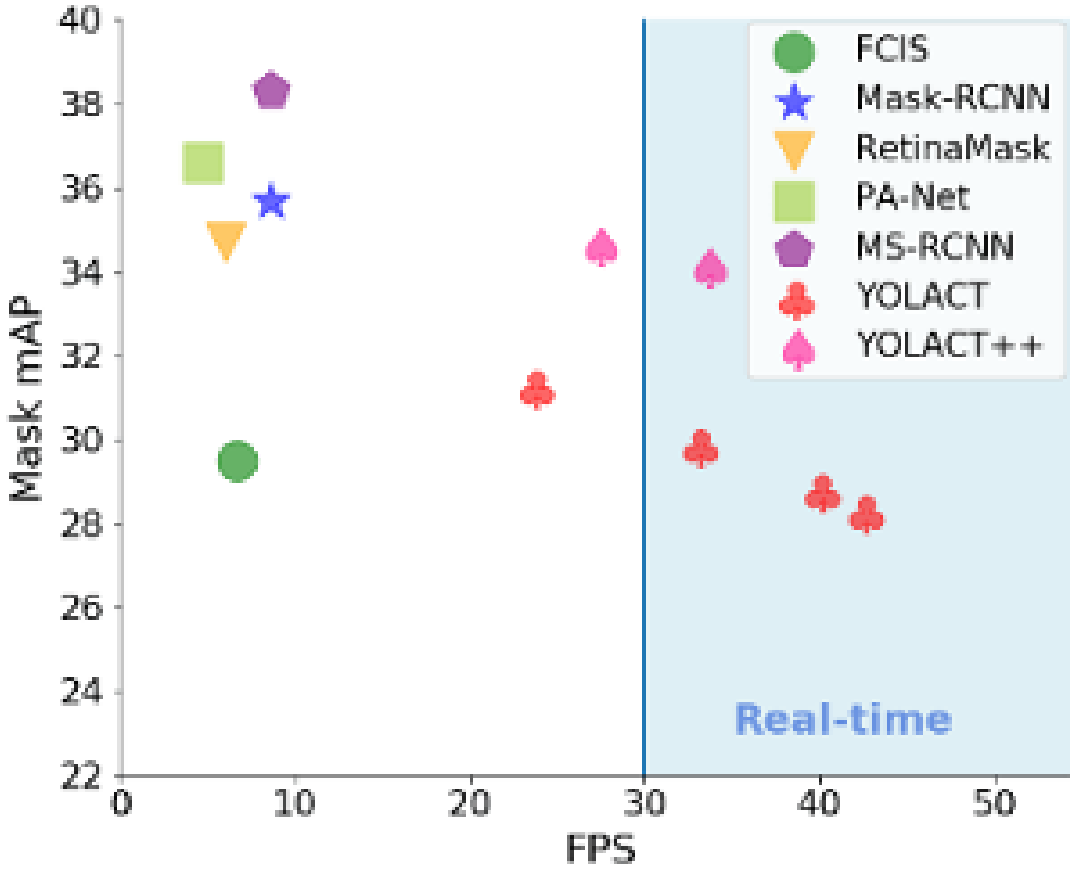


Figure 2.6: Instance Segmentation Models Comparison

2.3 Visual Object Tracker: Masking Single Player Across Sequential Frames

In sports analytics, Visual object tracking is essential for maintaining the identity of players as they move across the pitch. The precise masking of a single player across a sequence of frames in dynamic scenarios, such as football matches, is crucial for in-depth analysis.
The literature review explores a variety of methodologies, such as the correlation filters [4], Kalman filters [15], and more recent deep learning-based trackers like GOTURN [12] in addition to the work by Zhang et al. [35], providing insights into tracking and masking a moving player.

2.3.1 Mask R-CNN: Precise Segmentation Tracking

Mask R-CNN [10] stands out as a powerful methodology for instance segmentation, providing accurate masks for individual players in sports footage. Its simultaneous prediction of object bounding boxes and segmentation masks makes it particularly effective in isolating and tracking players in the dynamic context of football matches.

2.3.2 Online Adaptation for Dynamic Player Appearance

Jianglong Ye et al. [32] propose online adaptation techniques to address challenges posed by dynamic changes in player appearance. By continuously updating the initial mask based on evolving player characteristics, these adaptive approaches ensure the persistence of accurate player masks across frames, even when facing changing appearances.

2.3.3 Attention Mechanisms for Region Refinement

Vaswani et al. [29] inspire the use of attention mechanisms for refining the masked region around the player. These techniques focus on relevant regions within the initial mask, adapting it to the player's movement. This attention-driven refinement contributes to the precision of the player mask in dynamic sports scenarios.

2.4 Path Prediction in Sports Dynamics

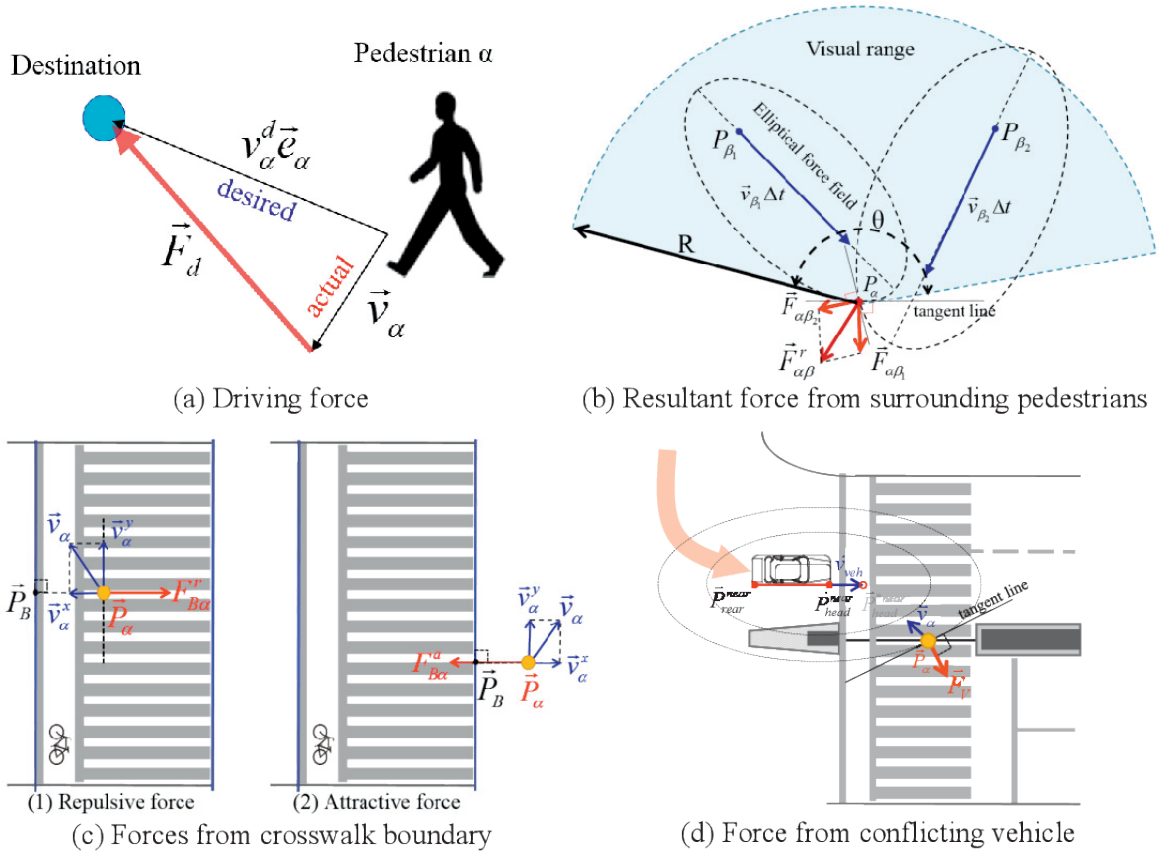
Path prediction aids in anticipating the future movements of players or the ball, contributing to occlusion resolution. Recurrent Neural Networks (RNNs) [26] and Long Short-Term Memory (LSTM) networks [13] are prevalent in trajectory prediction. It plays a crucial role in understanding the movement of entities, with applications ranging from crowd management to sports analytics. We explore the methodologies of knowledge-based (KB) and deep learning (DL) in the application of path prediction in sports scenarios.

In sports analytics, studies like [20] have applied path prediction to player movement analysis. Examining these works provides insights into the challenges and solutions related to path prediction in football scenarios.

2.4.1 KB Approaches

Traditional models, such as those based on social force principles [11] and cellular automata [6], contribute valuable insights into crowd dynamics. These models incorporate parameters for social interactions and environmental factors. While highly interpretable, KB models may face challenges in accommodating diverse pedestrian behaviors.

KB models find applications in sports analytics by applying collision avoidance and spatial anticipation principles. These models consider factors such as player positions, ball dynamics, and strategic elements. Challenges in adapting to dynamic sports scenarios may exist.



(a) Driving force (b) Resultant force from surrounding pedestrians

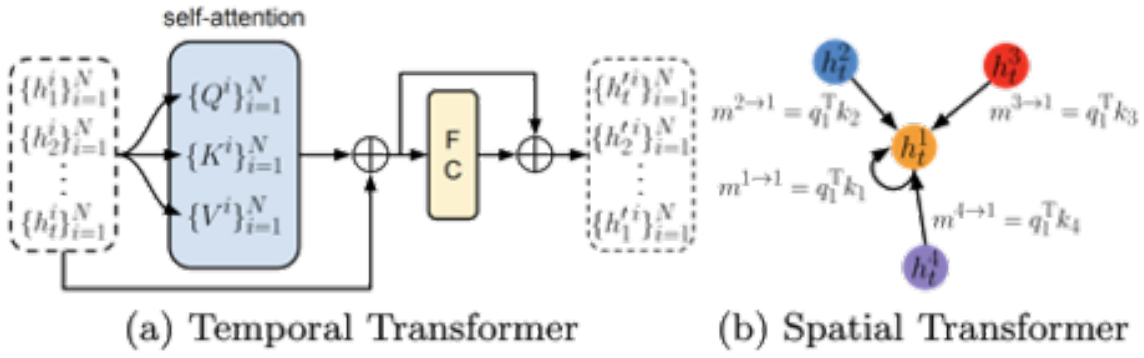
(c) Forces from crosswalk boundary (d) Force from conflicting vehicle

Fig. 1. Sources of social force at crosswalk

Figure 2.7: Social Force Model for Pedestrian Dynamics

2.4.2 DL Approaches

Deep Learning, specifically leveraging RNNs, LSTMs, and CNNs, has demonstrated remarkable efficacy in the intricate task of predicting pedestrian trajectories [2, 33]. RNNs and LSTMs excel in sequence modeling tasks due to their inherent ability to capture temporal dependencies within sequential data. LSTMs, with purpose-built memory cells, prove highly effective in various sequential prediction domains, such as speech recognition, machine translation, and trajectory forecasting [8, 28, 1, 30].



(a) Temporal Transformer (b) Spatial Transformer

Figure 2.8: Spatio-Temporal Graph Transformer Network

Despite the undeniable success of DL models, concerns linger regarding their interpretability and reproducibility, with the intricate nature of neural network architectures making it challenging to provide transparent insights into decision-making processes [21].

In the context of sports path prediction, DL approaches, particularly those utilizing recurrent and convolutional architectures, showcase significant promise. RNNs play a pivotal role in capturing temporal dependencies in sports trajectories, allowing models to discern intricate player movements, including passes, dribbles, and shots [34, 25]. The adaptability of DL models is particularly beneficial in understanding the nuanced and dynamic nature of sports scenarios. Concurrently, CNNs are adept at extracting spatial features from sports footage, facilitating the comprehension of player positioning on the field and interactions with teammates and opponents. The synergy of RNNs and CNNs in DL models empowers them to not only predict sports trajectories with high accuracy but also comprehend the complex interplay of temporal and spatial dynamics inherent in sports scenarios.

2.4.3 Challenges and Opportunities

The trade-offs between KB and DL approaches involve considerations of interpretability, reproducibility, and adaptability. Challenges, including data requirements, DL interpretability, and the need for domain knowledge in KB models, must be carefully navigated in the context of sports path prediction.
Path prediction in pedestrian dynamics and sports analytics presents a multidimensional challenge. Integrating KB and DL approaches provides a comprehensive perspective, leveraging their respective strengths for accurate and adaptable path predictions across diverse domains.

2.5 Similarity Matching

Similarity matching is critical for associating players across different poses, pictures, and camera viewpoints. The ability to maintain accurate player identification under occlusion scenarios is crucial for effective player tracking systems. This underscores the critical role of the similarity module, which facilitates re-identification of players after occlusion. Our proposed module utilizes Variational Autoencoders (VAEs) to encode each player into a unique latent vector before occlusion. Subsequently, we measure the dissimilarity between these latent vectors and those extracted from post-occlusion frames to establish matches.

To ensure the efficacy of this approach, we have conducted a comprehensive literature review encompassing two key areas:

- **Person Re-identification with Deep Learning:** We delve into existing deep learning-based re-identification approaches to glean insights into how they address occlusion challenges.

- **VAE for Molecular Similarity:** We explore the application of VAEs in characterizing molecular similarity, seeking transferable knowledge and potential adaptations for player re-identification under occlusion.

2.5.1 Person Re-identification with Deep Learning

Jun Xiang, Ziyuan Huang, Xiaoping Jiang, and Jianhua Hou [31] propose a novel deep learning conditional random field (Deep-CRF) graph for person re-identification, treating the task as a CRF node labeling problem. Their approach considers group-wise similarities within a batch of images and exhibits inference consistency in both training and testing stages. The Deep-CRF outperforms existing methods on three large-scale person re-ID datasets, surpassing the previous deep CRF framework by 8% in Rank1 accuracy on the CUHK03 dataset.

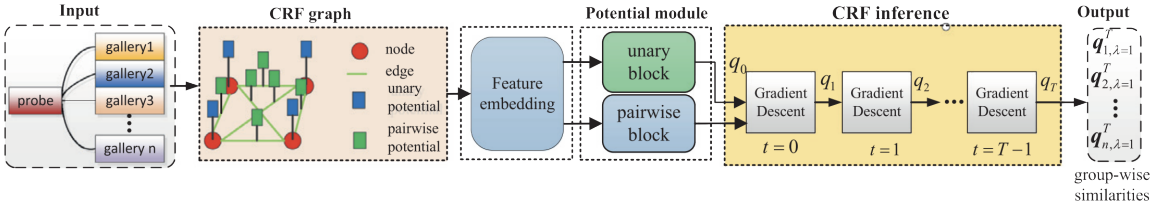


Figure 2.9: DeepCRF for Person Re-identification

Haque Ishfaq and Ruishan Liu [14] propose a novel structure named Triplet based Variational Autoencoder (TVAE) to enhance latent embedding learning by integrating deep metric learning with variational autoencoder (VAE). The TVAE method addresses the limitations of traditional VAEs in handling label or feature information, utilizing a triplet loss on the mean vectors of VAE in conjunction with reconstruction loss. Demonstrated on the MNIST dataset, TVAE achieves a high triplet accuracy of approximately 95.60%, showcasing its effectiveness, and the authors further validate the method on the Zappos50k shoe dataset, highlighting its efficacy in real-world applications.

2.5.2 VAE for Molecular Similarity

Soumitra Samanta, Steve O'Hagan, Neil Swainston, Timothy J. Roberts, and Douglas B. Kell [27] present VAE-Sim, a molecular similarity measure based on a variational autoencoder (VAE). The VAE is trained on over six million druglike molecules and natural products, utilizing a "bowtie"-shaped artificial neural network with a bottleneck layer for encoding and decoding. VAE-Sim offers a rapid and easily calculable metric for molecular similarity, providing a valuable contribution to cheminformatics.

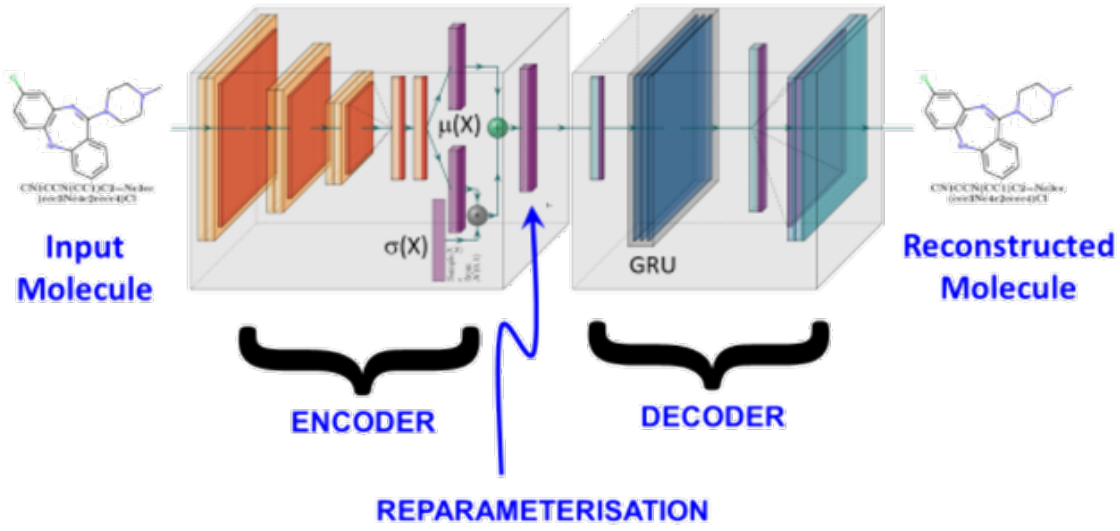


Figure 2.10: VAE-SIM Molecular Similarity

2.6 Future Research Directions

Robust Tracking Algorithms:

Investigating multi-object tracking approaches and incorporating contextual information (e.g., player interactions, field boundaries) could enhance robustness to occlusions and challenging conditions.

Efficient Model Architectures:

Exploring alternative model architectures or techniques for object detection, tracking, and path prediction could improve both accuracy and efficiency.

Real-Time Capabilities:

While not the current focus, research into real-time player tracking systems could expand the system's potential applications.

Multi-Camera Integration:

Developing strategies for seamless integration of multiple cameras, potentially with overlapping fields of view, could provide a more comprehensive and robust tracking experience.

Chapter 3

Neural Network Architecture

In this chapter, we provide a brief presentation of built neural networks is provided in the context of image and video processing. Inspired by the human brain, NNs are able to learn and generalize from experience. One major application area of a certain type of NNs is the convolutional neural networks (CNNs) and recurrent neural networks (RNNs). Both are advanced information processing paradigms inspired by the human brain, adapted specifically for tasks related to image and video analysis, such as image recognition, classification, segmentation, and temporal pattern recognition.

CNNs are specialized neural networks designed for processing grid-like data, such as images. Inspired by the human visual system, CNNs excel at capturing hierarchical features and patterns in images. Their ability to automatically learn and generalize from examples makes them particularly effective in tasks like image classification and object detection. CNNs are adept at discerning intricate relationships in data, even when the underlying structures are complex or challenging to articulate. Moreover, they demonstrate a remarkable capacity for extrapolating predictions to unseen aspects of the input data, enabling them to forecast future behaviors in image datasets.

RNNs, on the other hand, are tailored for handling sequential data, making them well-suited for video processing tasks. Inspired by the human brain's ability to recognize patterns over time, they possess a memory component that allows them to retain information about previous inputs. This makes them particularly effective in tasks such as video classification, where temporal dependencies play a crucial role. RNNs can learn and generalize from sequential examples, capturing nuanced relationships within video data even when the precise underlying connections are elusive. Their universal functional approximation capabilities empower RNNs to model complex nonlinear relationships without prior knowledge of the intricate interplay between input frames and output predictions.

In summary, CNNs and RNNs in image and video processing leverage their respective architectural strengths to learn, generalize, and forecast patterns within visual data, showcasing their adaptability and effectiveness in handling complex tasks inspired by human cognitive processes.

3.1 Object Detection Model

The chosen architecture for our object detection model is RetinaNet, a robust framework initialized using `torchvision.models.detection.retinanet_resnet50_fpn_v2`. It combines a ResNet-50 backbone with a Feature Pyramid Network (FPN) for effective multi-scale feature extraction.

The RetinaNet classification head has been customized to suit the specific requirements of the task:

- The number of anchors has been adjusted based on the model's default configuration.
- A group normalization layer with 32 groups has been introduced.
- The loss function is set to a custom Focal Loss, designed to handle class imbalance in object detection.

3.1.1 Loss Function

The Focal Loss employed in this model is tailored to address class imbalance in object detection. It includes tunable hyperparameters:

- `alpha` for balancing class weights.
- `gamma` for controlling the focus on hard-to-detect instances.

3.1.2 Model Configuration

During model instantiation using the `create_model` function, specific parameters are configured to tailor the model's behavior according to the task requirements.

- The model is designed to detect objects belonging to two classes: 'Player' and 'Ball.'
- Batch size is set to 1 for efficient GPU memory utilization.
- Base image resolution for transformations is set to 640.
- Image width and height are adjusted to 2688 and 1520, respectively.
- The model is trained for 75 epochs to capture diverse patterns in the data.
- Parallel data loading with 4 workers enhances training efficiency.
- The initial learning rate is set to 0.005 for effective weight updates.

Multi-Resolution Training

While our model primarily processes images at the base resolution, we explore the nuances of multi-resolution training and its implications. This allows us to understand the trade-offs associated with resolution variations during the training process.

Multi-resolution training is disabled (`RESOLUTIONS = None`). The model processes images at the base resolution and does not vary the resolution during training.

3.2 Instance Segmentation Model

This model is planned to be developed and deployed throughout the second semester Spring 2024 where we continue working on this project. The plan for the second semester is further discussed in the last chapter of this thesis.

3.3 Visual Tracking Model

The Visual Tracking Module has a pivotal role within the player tracking system, deploying an intricate Autoencoder architecture. This module is dedicated to the generation of precise monochromatic masks representing individual players, thereby enabling meticulous tracking across successive frames.
The masks generated by this module are fundamental inputs for subsequent phases of the player tracking process. These masks significantly contribute to the overall system's efficacy in precisely locating and tracking each player throughout the temporal evolution of a football matches.

3.3.1 Autoencoder Architecture

At the core of the Visual Tracking Module lies an Autoencoder neural network, distinguished by an encoder and a decoder, working in tandem to transform input images into informative player masks. The encoded information serves as a concise representation of the player, facilitating coherent tracking across frames.
The iterative refinement of the Autoencoder's architecture and training strategies underscores the module's adaptability to diverse player appearances and dynamic on-field scenarios. This continual improvement culminates in heightened precision and reliability within the player tracking subsystem.

Encoder

The encoder rigorously analyzes input images, encapsulating crucial features relevant to player attributes.

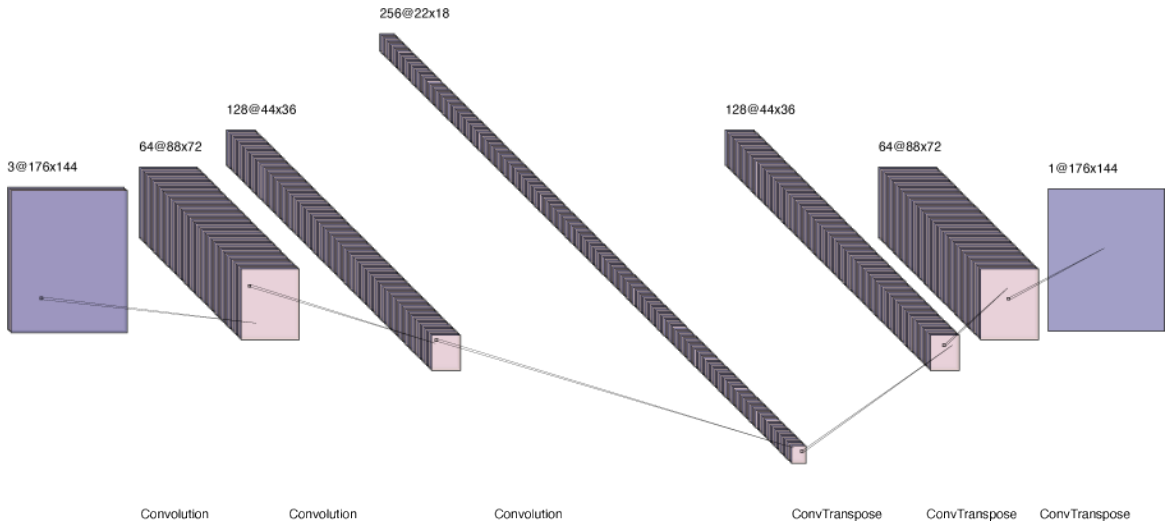


Figure 3.1: Autoencoder Architecture

- **Convolutional Layers:** The encoder comprises three convolutional layers strategically designed for hierarchical feature extraction from input images.
- **ReLU Activation:** Employing ReLU activation functions after each convolutional layer introduces non-linearity, enhancing feature representation.
- **Downsampling:** Strategic implementation of strided convolutions induces downsampling, simultaneously reducing spatial dimensions while increasing feature depth.

Decoder

The decoder intricately reconstructs input features into a monochromatic mask, highlighting the presence of players within the frame.

- **Transposed Convolutional Layers:** The decoder utilizes transposed convolutional layers to meticulously reconstruct spatial intricacies from the encoded features.
- **ReLU Activation:** Similar to the encoder, ReLU activation is judiciously applied to introduce non-linearity throughout the decoding process.
- **Sigmoid Activation:** The final layer employs a Sigmoid activation function, ensuring pixel values of the resultant mask fall within the [0, 1] range.

3.3.2 Initial Training Approach

The Visual Tracking Module is initially trained using a single image for each player and its corresponding mask, aimed at optimizing the Autoencoder's parameters for accurate player binary mask generation. The training process involves the utilization of Mean Squared Error (MSE) Loss as the optimization criterion to measure the difference between the predicted

and ground truth masks, coupled with the Adam optimizer to efficiently update the model parameters.

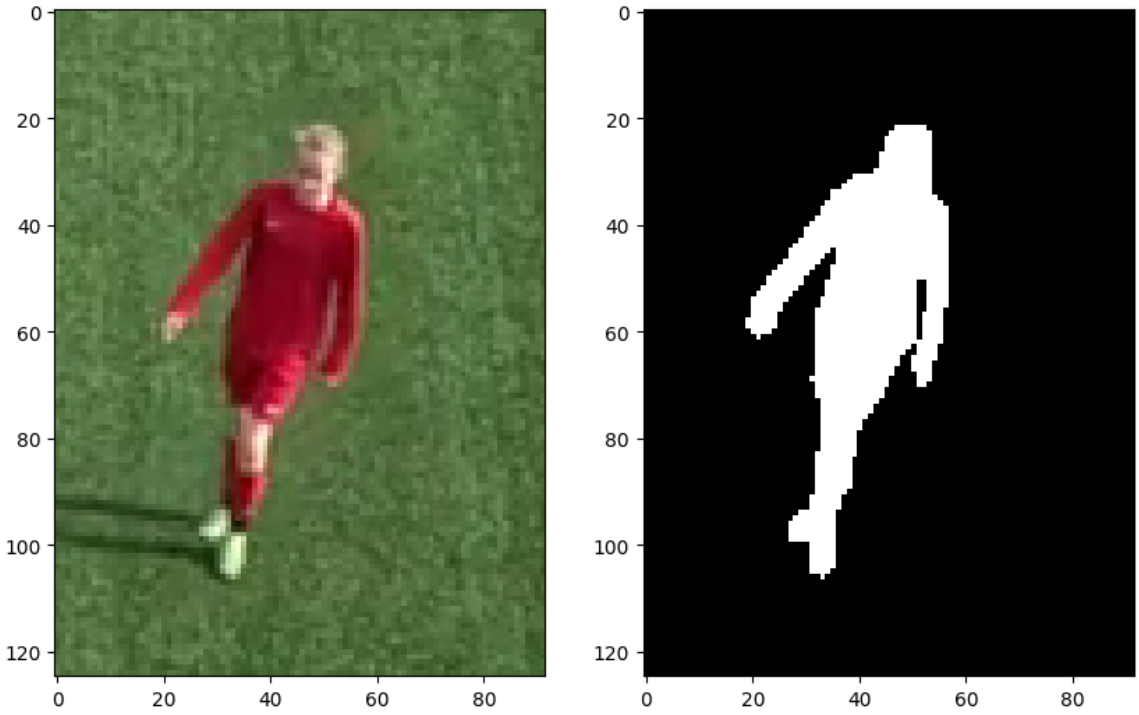


Figure 3.2: Autoencoder Training Input

MSE Loss

The Autoencoder training process involves minimizing the MSE loss function as an optimization criterion. This formula calculates the average of the squared differences between the actual and predicted values across all samples, providing a measure of the mean squared error between the predicted and true values.

$$\text{MSE Loss} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{3.1}$$

Where:

N represents the total number of samples.

y_i is the ground truth value for the i-th sample.

$\hat{y}_i$ is the predicted (model's output) value for the i-th sample.

3.4 Path Prediction Model

The path prediction module is needed for cost efficiency, as it allows the use of less expensive models when not needed. That is why path prediction is required to be able to detect if any occlusion (players covering each other from being viewed from the camera) will occur to

substitute models with each other.
Various attempts were made using knowledge-based systems like the Kalman Filter Predictor or using LSTM NN to be able to predict the upcoming movement of the player. However, these systems did not produce reliable enough accuracies to be useful. The system that did work on the other hand was one which utilized the UNET NN Architecture.

3.4.1 UNET Architecture

The name UNET comes from its U-shaped architecture which is famously used for segmentation. The left side of the U-shaped network consists of a series of convolutional and pooling layers. These layers serve as the encoder and are responsible for capturing the hierarchical features of the input image. At the bottom of the U, there is a bottleneck layer that connects the encoder. The right side of the U-shaped network is the decoder, which involves up-sampling and concatenation operations. The goal of the decoder is to generate a high-resolution segmentation map from the low-resolution feature maps obtained by the encoder. Skip connections connect the corresponding encoder and decoder layers which helps in retaining fine-grained details by allowing the network to use low-level features from the encoder during the up-sampling process in the decoder.

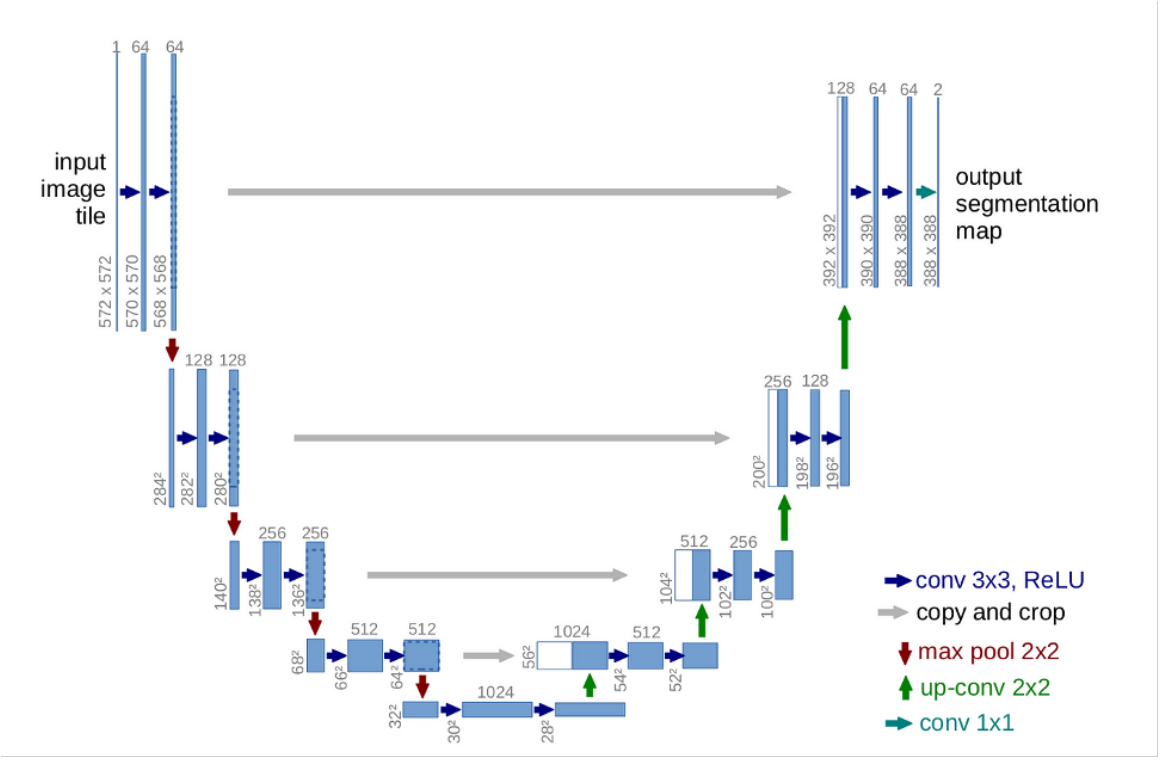


Figure 3.3: UNET Architecture

3.4.2 Model IO

Instead of using the UNET architecture to segment images, it was used instead to be trained on predicting player paths. This was done by providing binary masks of a player's bounding box of 25 frames (subject to change) to add to it his/her movement of the next 25 frames. Here's a sample of how the training data looks:

Figure 3.4: Path Prediction

Figure 3.5: Path Prediction Masks

3.5 Similarity Model

The Similarity Module stands as the unyielding choreographer of player tracking, expertly navigating occlusion's chaos through VAEs. It crafts unique latent spaces for each player, capturing their essence in mean and log variance vectors—digital fingerprints that persist even when players vanish from sight.

When occlusion strikes, the module compares these latent representations, re-identifying individuals and restoring order to the ballet of motion. This resilience stems from meticu-

lous training: each VAE dedicates itself to a single player, forging a deep understanding that transcends momentary disappearances.

3.5.1 VAE Architecture

This artistic network captures the player's essence through an encoder-decoder duet, forging latent fingerprints impervious to occlusion's veil. Its reparameterization magic brushstroke explores, finding familiar features beneath the surface, seamlessly reuniting players with their shadows. Watch the VAE weave a captivating tapestry of the match, even when players go dark.

Encoder

The encoder meticulously crafts a latent representation of the input image.

- Convolutional layers act as its brushstrokes, capturing intricate details and patterns.
- Pooling layers downsample, focusing on key features and forging a more abstract, condensed representation.
- Activation functions introduce non-linearity, enabling the model to capture complex relationships between visual elements.
- The final output: z_{mean} and z_{log_var} encapsulates the image's core essence, ready to be shaped into a reconstruction.

Decoder

The decoder accepts the latent vector as its muse, masterfully recreating the visual masterpiece from its distilled essence.

- Transposed convolutional layers, like a sculptor's deft hands, gradually add detail and structure, breathing life back into the image.
- The sigmoid activation acts as the final polish, ensuring a visually pleasing and coherent reconstruction, poised to captivate the viewer.

Reparameterization Trick

Traditional variational inference: directly sampling from a distribution during backpropagation, a process that's inherently non-differentiable.

- It introduces a subtle but powerful shift in perspective, enabling smooth gradients and efficient learning within continuous latent spaces.

- This element of chance, carefully blended with z_{mean} and z_{log_var}, creates a latent vector (z) that's both rooted in the input image's essence and infused with a touch of creative exploration.
- The resulting latent vector becomes the decoder's muse, guiding its brushstrokes towards a faithful yet nuanced reconstruction.

3.5.2 Model Training

In the training process of the provided VAE, the model is optimized using a combination of pixel-wise reconstruction loss (MSE) and Kullback-Leibler (KL) divergence loss. This dual-loss strategy is fundamental in VAEs for achieving two main objectives: accurate reconstruction of input data and regularization of the latent space.

Pixel-wise Reconstruction Loss (MSE Loss)

The pixel-wise reconstruction loss measures the dissimilarity between the input images and the images reconstructed by the decoder. It quantifies how well the VAE can reproduce the original data. The MSE is commonly employed as the reconstruction loss in VAEs. For each pixel in the input image, the MSE calculates the squared difference between the original pixel intensity and the corresponding pixel intensity in the reconstructed image. This is done for all pixels and averaged across the entire image. The goal is to minimize this loss, encouraging the VAE to generate reconstructions that closely resemble the input data.

$$\text{Loss}_{\text{pixelwise}} = \text{MSE Loss} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{3.2}$$

Where:

N is the total number of samples.

y_i is the ground truth value for the i-th sample.

$\hat{y}_i$ is the predicted (model's output) value for the i-th sample.

Kullback-Leibler (KL) Divergence Loss

The KL divergence loss is a regularization term that ensures the learned latent space follows a specific probability distribution, typically a unit Gaussian distribution. It penalizes deviations of the learned latent distribution from the chosen prior distribution (in this case, a Gaussian distribution with mean 0 and variance 1). Minimizing the KL divergence helps prevent overfitting and ensures that the encoder learns a meaningful and smooth representation of the input data.

The formula for the KL divergence loss in the context of a VAE is derived from the information theory concept and measures the difference between the learned distribution and the

target distribution. It encourages the latent space to be continuous and well-structured. Let's denote the distribution learned by the encoder as $Q(Z \mid X)$, where Z is the latent space and X is the input data. We want this distribution to be close to a specified prior distribution, usually a standard normal distribution P(Z).

$$\mathrm{D}_{KL}(Q(Z|X)||P(Z)) = \frac{1}{2}\sum_{i=1}^{N}(\mu_i^2 + \sigma_i^2 - \log(\sigma_i^2) - 1) \tag{3.3}$$

Where:

μ_i^2 is the squared mean of the learned distribution.
$\log(\sigma_i^2)$ is the log variance of the learned distribution $Q(Z \mid X)$.
-1 serves as a constant offset.

The goal during training is to minimize the KL Divergence term, encouraging the distribution $Q(Z \mid X)$ to approximate the desired prior distribution $P(Z)$. This encourages the latent space to be continuous and well-structured, making it more amenable to generating new, meaningful samples during the generative process.

Total Loss and Optimization:

The total loss for each training iteration is the sum of the pixel-wise reconstruction loss and the KL divergence loss, emphasizing both accurate reconstruction and regularization of the latent space.

$$\mathrm{Loss}_{\mathrm{total}} = \mathrm{Loss}_{\mathrm{KL}} + \mathrm{Loss}_{\mathrm{pixelwise}} \tag{3.4}$$

The model parameters are updated to minimize this combined loss using the Adam optimizer.

Chapter 4

Solution Architecture

4.1 System Overview

The football analytics system is designed to revolutionize player tracking and analysis in the dynamic world of football. Our system is a sophisticated blend of cutting-edge technology and a deep understanding of the sport, aiming to provide comprehensive insights into player movements and team dynamics.

This chapter delves deep into the intricate architecture of our multi-module player tracking system, designed to dominate the dynamic world of football matches. The system orchestrates five specialized modules in a distributed dance, each contributing unique skills to achieve robust and accurate player tracking.

At its core, the system seeks to automate and enhance the traditional manual methods of tracking player movements during football matches. The purpose is to provide accurate positional data for every player and the ball at every moment, across multiple camera feeds. This data forms the basis for in-depth player and team performance analysis.

The system's functionality is driven by the integration of deep learning advancements with other computer vision techniques. By harnessing the power of artificial intelligence, our goal is to overcome the limitations of manual tracking, such as labor-intensive processes, limited granularity, and occlusion issues.

Goals

Our primary goals include:

- Develop a modular system with specialized modules for player tracking, object detection, path prediction, instance segmentation, and similarity analysis.
- Provide accurate and detailed player statistics for performance assessment.
- Create a dynamic user interface for presenting the output statistics in an intuitive and graphically rich format.

- Ensure the system's efficiency in tracking all players and the ball, even in fast-paced scenarios.
- Facilitate human-agent interaction for updating player identification in cases of uncertainty.

Through this system, we aim to redefine how football analytics is conducted, offering a valuable and smart toolset for coaches, analysts, and football enthusiasts to gain deeper insights into the beautiful game.

4.2 A Modular Architectural Design

The system architecture is designed to seamlessly integrate multiple modules, rather than relying on a single conductor each specializing in a crucial aspect of the tracking process. The architecture is modular, allowing for flexibility, scalability, and efficient communication between components.

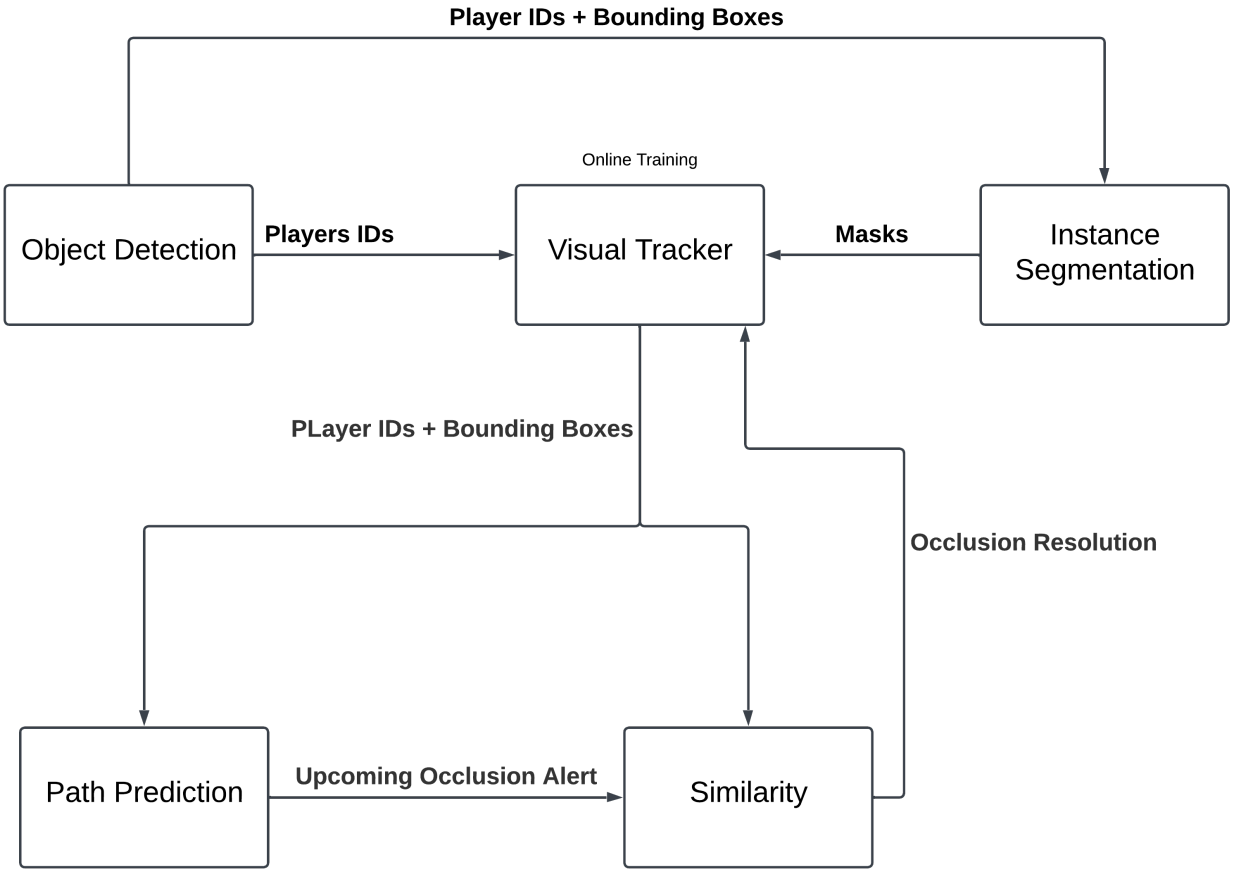


Figure 4.1: Software System Architecture

The modules communicate seamlessly to ensure the flow of data across the system. A high-level communication diagram is presented in Figure 4.1. The modular architecture ensures that

each component can be individually upgraded or replaced without affecting the entire system. This flexibility allows for the incorporation of advanced technologies and methodologies as they emerge in the field of football analytics. Five modules, each a virtuoso in its own right, work together in seamless harmony.

4.2.1 Object Detection Module

Within our player tracking symphony, the object detection module acts as a spotlight operator, precisely illuminating the players and the ball on the pitch. It wields ResNet50 as its discerning lens, identifying these key objects and outlining their positions within the scene. When new players enter the stage, this module steps up, using the sharp eyes of ResNet50, trained on 10,000 football match labeled frames. It assigns unique IDs and generates initial masks, setting the scene for the other modules to shine. It identifies and locates objects such as players and the ball within the camera feeds.

Detection and Localization:

- Receives video frames from the system's input pipeline.
- Employs ResNet50 to detect players and the ball within each frame.
- Generates bounding boxes surrounding each detected player and ball.
- Assigns unique IDs to each detected player.

Module Output

Provides the following information to subsequent modules:

- Bounding boxes for each detected player and the ball, specifying their coordinates and dimensions.
- Unique player IDs, enabling identification and tracking.

Activation:

The object detection module is called upon for initial identification of players and the ball at the start of the analysis process. In addition, It is engaged when new players enter the field of view or when occlusions require re-identification. It also collaborates with the visual tracker module to maintain tracking.

Key Considerations:

- **Accuracy:** Precise detection and localization are essential for accurate tracking and analysis.

- **Speed:** Real-time processing demands efficient inference.
- **Robustness to Occlusions:** Handling partial visibility is crucial for maintaining tracking integrity.
- **Integration with Other Modules:** Seamless collaboration with the visual tracker, instance segmentation, and similarity modules is vital for comprehensive player tracking.

Future Enhancements

- Exploration of alternative object detection models for potential performance gains.
- Incorporating contextual information (e.g., field lines, player interactions) to improve accuracy.
- Investigating techniques to enhance robustness to challenging lighting conditions.

4.2.2 Visual Tracker Module: Persistent Player Trajectory Analysis

Within our player analysis symphony, the visual tracker module plays the role of a persistent and steadfast conductor, orchestrating the spotlight monitoring on each player's movements across the pitch. The module processes subsequent video frames, employing a multi-stage approach where it wields the intricate language of autoencoders, connected components analysis, and moving window algorithms to maintain a continuous understanding of player trajectories.

Functionality

1. **Initializing Player Representations and Activation:**
    - Initiated after initial player identification by the object detection module.
    - Sustains continuous operation throughout the analysis process to maintain tracking.
    - Receives initial bounding boxes and masks from the object detection module.
    - Initialize a moving window for every player using the given bounding box, where:

$$\begin{aligned} \text{width of the window} &= 2 \times \text{bbox width}, \\ \text{height of the window} &= 2 \times \text{bbox height}, \\ \text{center of the window} &= \text{center of the bbox}. \end{aligned}$$

    - Implements autoencoder training for each player, facilitating the acquisition of compact representations of their visual attributes.
2. **Autoencoder Feedforward:**

- Applies the trained autoencoders to generate masks for each tracked player to generate binary (black and white) masks of the given window.

3. **Connected Component Analysis:**

   - Applies connected components analysis to discern individual players within the generated masks, and draw bbox on the edges of the largest connected component in the window mask.

4. **Update Moving Window:**

   - Employs a moving window algorithm to dynamically follow players across frames, to minimize the processing time and effort needed where the autoencoder works on a fraction of the frame instead of processing the full frame.
   - The updated window center point is the center point of the new bbox, and the updated width and height are double the width and height of the new bbox.
   - This moving window algorithm is scale invariant as it adapts to changing sizes of the players as they move closer or further from the camera.

5. **Online Adaptation:**

   - Executes periodic updates to the DL model through continuous training of the autoencoders (every 50 frames) to refine player representations and adapt to variations in appearance, lighting, or camera angles.

6. **Module Output:**

   - Provides updated bounding boxes and masks for each tracked player, reflecting their positions and movements in each frame.

Key Considerations

- **Accuracy:** Precise tracking of player trajectories is essential for reliable analysis of player behavior and interactions.
- **Robustness to Occlusions:** Handling partial visibility and overlapping players effectively is crucial for maintaining accurate tracks.
- **Efficiency:** While not real-time, the visual tracker module utilizes an efficient tracking algorithms which is super important to minimize processing time.
- **Adaptability:** The ability to adapt to appearance changes and variations in camera views is crucial for maintaining tracking integrity.

Future Enhancements

- Exploration of alternative autoencoder architectures or tracking algorithms for potential performance gains.
- Incorporating contextual information (e.g., player interactions, field boundaries) to improve tracking accuracy particularly to handle occlusions more effectively.
- Investigating techniques to enhance robustness to challenging lighting conditions and abrupt appearance changes.
- Researching multi-object tracking approaches that leverage contextual relationships between players for improved overall performance.

4.2.3 Path Prediction Module: Foreseeing Occlusions, Optimizing Efficiency

Within our player tracking orchestra, the path prediction module plays the role of a cautious prophet, peering into the future to anticipate occlusions and guide the system towards cost-effective tracking. It utilizes the powerful UNET architecture, not for traditional segmentation, but as a seer of player trajectories, enabling strategic resource allocation. By predicting future trajectories, it warns the other modules of potential occlusions, allowing for smoother transitions and accurate tracking.

By utilizing the path prediction module's foresight, the system can navigate the cost-efficiency terrain effectively, allocating resources wisely while maintaining accurate player tracking throughout the analysis.

Functionality

1. **Occlusion Foresight:**
    - Analyzes player bounding boxes for a specified time window (e.g., 25 frames).
    - Employs the trained UNET model to predict potential occlusions based on player movements and relative positions.
    - Calculates the probability of occlusion occurring for each player pair.
2. **Cost-efficient Resource Allocation:**
    - Leverages occlusion predictions to determine the appropriate tracking model for each player in the next frame.
    - Substitutes computationally expensive models with simpler, yet sufficient, alternatives for players unlikely to be occluded.
    - This dynamic switching optimizes processing time and resource consumption while maintaining accurate tracking.

Module Output

- Occlusion probability scores for each player pair.

Activation

- Operates concurrently with the visual tracker module, analyzing input data every frame.
- Predictions inform model selection choices before tracking updates are performed.

Key Considerations

- **Prediction Accuracy:** Reliable prediction of potential occlusions is crucial for effective resource allocation.
- **Cost Optimization:** Balancing tracking accuracy with computational efficiency is key for maximizing cost-effectiveness.
- **Integration with Other Modules:** Seamless collaboration with the visual tracker and instance segmentation modules is vital for accurate tracking even when occlusions occur.

Future Enhancements

- Incorporating additional contextual information (e.g., ball position, team tactics) to refine occlusion predictions.
- Investigating reinforcement learning approaches to dynamically adjust model selection based on real-time performance and resource constraints.
- Researching integration with the similarity module to leverage cross-camera player information for improved prediction accuracy.

4.2.4 Similarity Module: Dynamic Player Re-Identification Framework

A bridge across time and space, this module recognizes players even when their appearance changes due to occlusions or camera handoffs. Using the art of a variational autoencoder and KL divergence, it matches players across these gaps, ensuring continuous tracking throughout the match. It analyzes similarity between player looks, helping the re-identification process using similarities.

In the realm of player tracking, ensuring robust identification of individuals under occlusion remains a formidable challenge. Addressing this challenge necessitates the development of innovative solutions that can effectively re-identify players before and after occlusion. This section introduces a novel similarity module designed to leverage Variational Autoencoders (VAEs) for the purpose of re-identifying players in the dynamic context of occlusion scenarios.

Module Functionality

The functionality of the proposed similarity module is centered on training individual VAE models for each player before occlusion. When confronted with occluded players, the module passes their visual data through all pretrained models. The resulting mean μ and log variance $\log(\sigma_i^2)$ vectors serve as latent representations for comparison.

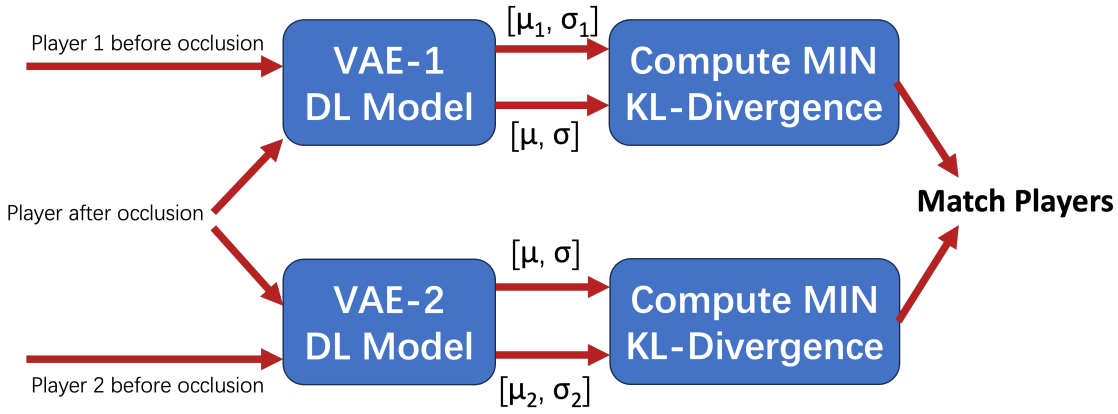


Figure 4.2: Similarity Module Operation

VAE Feedforward

In the feedforward process, the VAE encodes an input image through its convolutional layers, leading to the extraction of high-level features. The mean and log variance vectors are then computed from the flattened representation. Subsequently, during the decoding phase, these latent vectors are used to reconstruct the original image. This dual process enables the module to capture intricate patterns while facilitating the generation of data.

Comparing Players Before and After Occlusion

A set of pre-occlusion cropped frames is saved for each player, subsequently utilized to train individual Variational Autoencoder (VAE) models, capturing distinctive features expressed through mean and logarithmic variance representations. Following the occlusion period, wherein players become spatially separated, cropped frames are extracted for each player. These frames undergo processing through all pre-trained occluded player models to establish a match between the post-occlusion player image and the pre-occlusion player frames. This matching process unfolds through a meticulously orchestrated sequence of steps:

1. Forward the post-occlusion player image through the respective model, extracting mean and log variance representations.

2. Compute mean and log variance values for the stored frames corresponding to the utilized model.

3. Employ the Kullback-Leibler (KL) divergence metric to ascertain the minimum dissimilarity between the mean and log variance extracted from the post-occlusion player image and the corresponding values derived from the saved pre-occlusion frames.

4. Iteratively execute the previous steps for all player models encompassed in the occlusion scenario. Identify the minimum dissimilarity among all calculated minimums as the conclusive player identification.

4.2.5 Instance Segmentation Module

A master of disguise, this module intervenes during occlusions, wielding its specialized tools to segment individual players from the entangled mass. Keeping the identities clear, it ensures the tracking melody doesn't lose its rhythm. It segments individual instances within a scene, providing detailed information about each player and the ball to be able to re-identify a player correctly after an occlusion event.

This module is planned to be developed and deployed throughout the second semester Spring 2024 where we continue working on this project. The plan for the second semester is further discussed in the last chapter of this thesis.

4.3 Integration and Data Flow

The efficacy of our player tracking system is contingent upon the orchestrated collaboration of its constituent modules, intricately interweaving information through a systematically designed data flow. This section delves into the technical underpinnings of the integration mechanisms and data exchange protocols that govern the seamless functioning of these modules, revealing the meticulous ballet orchestrating precise player analysis.

This meticulously designed architecture that will be integrated throughout the second semester serves as the foundational framework for our player tracking system, affording a cohesive and efficient analysis of football matches. The technical intricacies of data flow and module interactions illuminate the precision with which the system extracts accurate and valuable insights from the dynamic realm of football.

4.3.1 Data Flow

Input Pipeline:

- Video frames from fixed cameras constitute the raw input for analysis.

- Pre-processing steps, including noise reduction and color space conversion, refine the data for module consumption.

Player Detection:

- Processes pre-processed frames using the ResNet50 model.
- Output: Bounding boxes and assigned IDs for detected players and the ball.

Parallel Visual Tracking:

- Initiates with initial player information from the object detection module.
- Analyzes subsequent frames utilizing trained autoencoders, connected components analysis, and moving window algorithms.
- Output: Updated bounding boxes and masks for each tracked player.

Trajectory Prediction:

- Operates concurrently with the visual tracker, receiving player bounding boxes.
- Utilizes the UNET model to predict potential occlusions and recommends tracking model choices for the next frame.
- Output: Occlusion probability scores for player pairs.

Instance Segmentation:

- Activated by the visual tracker or path prediction module during occlusions.
- Employs a specialized algorithm to segment individual players within overlapping regions.
- Output: Refined accurate masks for each occluded player.

Similarity Matching:

- Activated during occlusions or camera handoffs to maintain player identity.
- Utilizes VAE and KL divergence to match players across gaps.
- Output: Verified player IDs for matched players across occlusions or camera views.

4.3.2 Module Interactions

- The object detection module initiates the workflow, identifying players and providing foundational information for subsequent modules.
- The visual tracker continuously monitors players, utilizing path prediction to anticipate occlusions and dynamically adjust resource allocation.

- Instance segmentation intervenes during occlusions, disentangling overlapping regions and contributing to the preservation of accurate tracking.
- The similarity module bridges gaps, ensuring consistent player identification across occlusions and camera handoffs.

4.3.3 Integration Mechanisms

- Shared memory buffers facilitate efficient data transfer between modules.
- Message queues enable asynchronous communication, promoting parallel processing.
- A central controller module oversees the overall data flow and orchestrates module interactions.

4.3.4 Benefits of Integrated Architecture

- Seamless collaboration ensures efficient and accurate player tracking.
- The system balances accuracy with computational costs, switching between multiple modules for nearly similar goals such as the visual tracking, instance segmentation, and object detection modules.

  Where the object detection and instance segmentation provide the highest fidelity when comes to the mask identification of the players but they come with major computational demand over that needed by the much efficient visual tracking module at the cost of medium mask accuracy.
- Path prediction optimizes resource allocation, striking a balance between accuracy and computational efficiency.
- Individual modules address specific tasks, promoting a modular and maintainable system.
- The system dynamically adapts to evolving game situations through continuous learning and data integration.

4.4 Software and Frameworks

The modular architecture ensures that each component can be individually upgraded or replaced without affecting the entire system. This flexibility allows for the incorporation of advanced technologies and methodologies as they emerge in the field of football analytics.

Our system finds its home on Linux-based operating systems, currently waltzing to the rhythm of Python, PyTorch, and OpenCV. However, a grand migration awaits, where C++

joins the band with LibTorch and OpenCV, enabling parallel processing through multithreading and multiprocessing. This orchestrated hardware-software harmony ensures efficient performance, essential for keeping up with the fast-paced ballet of football. Key software components and frameworks include:

- **Operating System:** The system is developed and tested on Linux-based operating systems to ensure compatibility and stability.
- **Deep Learning Frameworks:** PyTorch is the primary framework for implementing and training deep learning models due to its flexibility and robust tools.
- **Computer Vision Libraries:** OpenCV is utilized for image processing tasks, offering essential functionalities for video analysis.

Development Tools

Development is facilitated by the following tools:

- **Integrated Development Environment (IDE):** Visual Studio Code is used as the primary IDE for coding and debugging.
- **Version Control:** Git is employed for version control, supporting collaborative development and codebase management.

These technologies form the foundation of the football analytics system in the build, emphasizing high-performance computing and efficient processing of video inputs.

4.5 Limitations: Acknowledging the Boundaries

Even the most skilled players encounter challenges on the field. In this spirit, we acknowledge the limitations of our current player tracking system, identifying areas where further development could enhance its performance.

By acknowledging these limitations, we pave the way for future research and development, striving to create player tracking systems that can perform even more impressively on the challenging pitch of football analysis.

4.5.1 Challenging Conditions

- **Occlusions:** While the system employs strategies to handle overlapping players, performance can degrade in scenarios with severe or prolonged occlusions.
- **Lighting Variations:** Abrupt changes in lighting conditions, such as shadows or strong sunlight, can impact object detection and tracking accuracy.

- **Camera Views:** The two fixed cameras provide limited perspectives, potentially hindering tracking accuracy in certain areas of the pitch.

4.5.2 Computational Efficiency

- **Processing Time:** While not designed for real-time performance, the system's processing time could be further optimized for faster analysis, especially when handling high-quality videos.
- **Resource Allocation:** The path prediction module strives for cost-efficiency, but further improvements in model selection and resource management could enhance overall performance.

4.5.3 Accuracy Trade-offs

- **Balancing Accuracy and Efficiency:** The system balances accuracy with computational costs, but there remains potential to explore methods that further improve accuracy without sacrificing efficiency.
- **Instance Segmentation:** The specific algorithm used for instance segmentation may have limitations in handling complex occlusions or diverse player appearances.

4.5.4 Data and Model Limitations

- **Training Data:** The quality and diversity of training data directly impact model performance. Expanding the dataset with varied scenarios and player appearances could enhance robustness.
- **Model Generalization:** The system's effectiveness may vary across different camera setups, lighting conditions, and player behaviors. Exploring techniques to improve model generalization is crucial for broader applicability.

4.5.5 Cross-Camera Tracking

Camera Handoffs: The similarity module facilitates tracking across cameras, but accuracy could be improved by incorporating more sophisticated feature matching techniques or leveraging geometric constraints.

Chapter 5

Software Requirements Specification (SRS)

The Software Requirements Specification (SRS) outlines the functional and non-functional requirements of the proposed deep learning and computer vision system, emphasizing efficiency in tracking all players and the ball to generate detailed and in-depth statistics. The system aims to provide an intuitive and graphically rich user interface for presenting player statistics dynamically.

5.1 Scope

The scope of the system encompasses the development of a sophisticated efficient computer vision application that utilizes deep learning models for player tracking and in-depth analysis in football matches. The system aims to address the limitations of manual tracking methods and provide accurate positional data for every player, at every moment, across multiple camera feeds. Key functionalities include:

1. **Efficient Player Tracking:** The system must efficiently track each player on his own in addition to the ball throughout a football match.

2. **Occlusion Resolution Strategy:** The system should attempt to undergo occlusion resolution when players overlap and ask for human intervention under certain certainty threshold.

3. **In-Depth Player Statistics:** The system will generate comprehensive player statistics, capturing various aspects and statistics to represent and analyse each player performance in numbers.

4. **In-Depth Team Statistics:** The system will generate comprehensive team statistics, capturing various aspects and statistics to represent and analyse collective team performance in numbers.

5. **Dynamic User Interface:** The user interface will be intuitive and dynamic, providing an intuitive and graphically rich environment for users to interact with deep learning-based player tracking data.

6. **Human-Agent Interaction:** The system may require human-agent interaction in rare cases to update player identification when certainty falls below acceptable measures.

The system's development will adhere to specified budget and time constraints, focusing on efficiency and detailed accurate statistical analysis.

5.2 Functional Requirements

Player Detection and Identification

1. The system shall utilize a collection of state-of-the-art deep learning models for accurate player detection, and identification on the football field at all times.

2. It shall employ a neural network model to distinguish between each player on the pitch.

3. The deep learning models shall be trained to handle variations in player appearance, including different team jerseys and lighting conditions.

Player Tracking

1. The system shall implement deep learning-based tracking algorithms for continuous monitoring of each player throughout the match.

2. It shall leverage convolutional (CNNs) and recurrent neural networks (RNNs) or similar architectures to handle occlusion scenarios, providing robust tracking even in obstructed views.

3. The tracking models shall be adaptable to sudden changes in player speed and direction.

Integration with Deep Learning Models

1. The system shall seamlessly integrate deep learning models with other computer vision techniques for comprehensive player tracking.

2. It shall synchronize information from multiple camera feeds and combine features extracted by deep learning models for a unified view of player movements across the pitch.

User Interface

1. The system shall provide an intuitive user interface for coaches, analysts, and users to interact with deep learning-based player tracking data.
2. Users shall have the ability to customize views, select specific players, and replay specific game moments enriched with insights from deep learning analysis.

5.3 Use Cases

Use Case 1: Real-time Player Tracking

Description: The system shall efficiently and continuously track all players and the ball during a football match, providing accurate positional data.

Actors: Deep Learning Model, Camera Feeds

Flow:

1. The system receives video feeds from multiple cameras.
2. The deep learning model processes the video frames, detecting and identifying players.
3. The tracking algorithm updates the positional data of each player and the ball.
4. In rare cases where certainty falls below acceptable measures, the system alerts a human agent for player identification updates.
5. The system provides the updated player positions to the user interface.

Exception: The system shall handle unexpected interruptions in video feeds gracefully, resuming tracking upon feed restoration.

Use Case 2: Detailed Player and Team Statistics

Description: Users (coaches, analysts, and general users) can interact with the system to access player and team tracking data and in-depth statistics.

Actors: Statistical Analysis Module, User Interface, User.

Flow:

1. The system captures player movements and interactions with the ball.
2. The statistical analysis module processes the captured data to generate detailed player and team statistics.
3. The user interface dynamically presents the generated statistics to users.
4. The user accesses the system through the user interface.

5. The user customizes views, selects specific players, and explores historical data.

6. The system retrieves and displays the requested player tracking information.

Exception: If there are connectivity issues, the system shall provide a user-friendly error message to notify the user and attempt reconnection.

5.4 Non-Functional Requirements

Non-functional requirements define the attributes that characterize the system's operation and performance. These aspects are crucial for ensuring the system's effectiveness, reliability, and user satisfaction.

Performance

Efficiency

The system must efficiently process and analyze player and ball tracking data, providing near-real-time results during and after a football match.

Scalability

The system should be scalable to accommodate an increasing number of players and diverse game scenarios without significant degradation in performance.

Reliability

Availability

The system should strive to maintain high availability, minimizing downtime and disruptions during football matches.

Error Handling

The system must implement robust error-handling mechanisms to gracefully manage unexpected errors and ensure data accuracy.

Usability

User Interface

The user interface must be intuitive, responsive, and graphically rich, ensuring a positive and engaging user experience.

Accessibility

The system should adhere to accessibility standards, ensuring that users with diverse needs can interact with the interface effectively.

Security

Data Confidentiality

Player and team data collected and processed by the system must be kept confidential and protected against unauthorized access.

Integrity

The system must maintain the integrity of tracking data, preventing unauthorized modifications or tampering.

Compatibility

Hardware Compatibility

The system should be compatible with standard hardware configurations, ensuring broad accessibility.

Software Compatibility

The system must be compatible with common operating systems and software environments, facilitating easy integration.

Performance Metrics

Tracking Accuracy

The accuracy of player and ball tracking must meet or exceed predefined benchmarks, ensuring the reliability of generated statistics.

Response Time

The system's response time for user interactions and data processing should adhere to acceptable standards, providing a seamless experience.

5.5 Constraints

Constraints play a crucial role in shaping the boundaries and capabilities of the system. Understanding and addressing these constraints is vital for the successful development and deployment of the computer vision and deep learning-based football player tracking system.

Technological Constraints

PyTorch Framework Compatibility

The system's development is currently constrained by the compatibility of the PyTorch deep learning framework. The selected framework must effectively support the development and integration of advanced models for player and ball tracking. A planned migration to LibTorch in C++ during the second semester introduces an additional constraint for seamless transition and compatibility.

Computational Power

The efficiency of the system depends on the available computational resources. To ensure optimal performance, the system must be designed to leverage the computational power, GPU capabilities, and memory of the underlying hardware.

Human-Agent Interaction

In rare instances where system certainty is below acceptable measures, human-agent interaction may be required for player identification updates. Balancing autonomy with human intervention is a key operational constraint.

Regulatory Constraints

Data Privacy Compliance

The system must strictly adhere to data privacy regulations, ensuring the secure and compliant handling of player and team data. Compliance with regional and international data protection laws is mandatory.

Ethical Use of Data

Development and operation must align with ethical standards, emphasizing the responsible and respectful use of data. Ethical considerations involve protecting the privacy and rights of individuals involved in football matches.

Time Constraints

Semester-based Development Phases

The development and deployment of the system are structured within two distinct semesters. The first semester, spanning from October 2023 to January 2024, focuses on initial design, model development, and system implementation using PyTorch. The second semester, from February 2024 to June 2024, marks the planned migration to LibTorch in C++ and the refinement of the system based on insights gained from the initial phase.
The subdivision of the development process across semesters and the identification of time-specific milestones ensure a structured and well-paced approach to the project.

Project Timeline Milestones

The project timeline is delineated by specific milestones and deadlines. Adherence to these temporal markers is critical for the successful completion of each development phase. Key checkpoints include model training completion, system integration, and the finalization of the user interface.

Post-Match Analysis Turnaround

Given the duration of football matches, the system must efficiently conduct post-match analysis within a reasonable timeframe. Timely generation of comprehensive player and team statistics is essential for providing relevant insights to users.

Chapter 6

Prototypes Results

The development of an innovative sports analytics system necessitates a thorough examination of its constituent modules before their integration into a seamless framework. In this chapter, we present the results of individual prototypes for each module within our proposed system. The primary objective of these prototypes is to assess the performance, accuracy, and limitations of each module in isolation.

Our approach involves breaking down the system into five distinct modules, each catering to a specific aspect of sports analytics: Player Tracking, Object Detection, Path Prediction, Instance Segmentation, and Similarity Analysis. By isolating these modules during the prototyping phase, we aim to gain insights into their independent functionalities, strengths, and areas requiring refinement.

The significance of this stage lies in the granularity of analysis it provides. Evaluating each module independently allows us to delve into specific challenges, optimizations, and intricacies associated with player tracking and analysis. The results obtained from these prototypes not only serve as benchmarks for individual module performance but also inform the integration process, guiding us towards a cohesive and robust sports analytics system.

This chapter unfolds with detailed accounts of the outcomes from each prototype, offering a comprehensive understanding of the strengths and limitations inherent to each module. As we navigate through the results, we lay the groundwork for the subsequent chapter, where we explore the integration of these modules to create a holistic sports analytics solution.

6.1 Object Detection Model Refinement

The initial prototype of our object detection model revealed suboptimal performance, largely attributed to inaccuracies in image annotations. Recognizing the pivotal role of precise annotations in shaping the model's understanding, we undertook a thorough reassessment of our annotation strategy to enhance the interpretative capacity of the system.

6.1.1 Enhancements in Annotation Approach

The primary catalyst for significant improvements in model performance stemmed from strategic adjustments in our annotation approach. Key enhancements include:

Comprehensive Annotation of Balls

In the initial phases, our annotation strategy was limited to the match ball, potentially neglecting other instances of balls within the images. Understanding the importance of capturing every ball in the field of view, we refined our annotation protocol to comprehensively label all instances. This expansion ensured a more nuanced understanding of ball-related entities.

Inclusive Annotation of Human Entities

To account for the diverse entities present on the field, including players, referees, and any other personnel, we expanded our annotation criteria. Initially, our focus might have been solely on players, leading to oversight in crucial elements. The revised approach ensures that any human presence within the field is accurately annotated.

These adjustments were pivotal in refining the dataset quality, enabling the model to learn from a more comprehensive set of annotations. As a result, the prototype's performance experienced a significant boost, demonstrating improved accuracy and reliability in object detection.

Iterative Refinement Process

The iterative process of refining annotations underscores the significance of meticulous data preparation in the success of the object detection model. Through continuous refinement and attention to detail, we successfully mitigated the limitations observed in the initial prototype, laying the groundwork for more promising results in subsequent evaluations.

6.1.2 Results Samples

Despite these improvements, challenges persisted, as illustrated in the figure below. Issues such as incorrect ball handling and misidentification of players remained, prompting further investigation and refinement in subsequent phases.

Failed Results: As shown below in the figure, the ball is not correctly handled and not all players detected are actually players.

Figure 6.1: Failed Sample of Object Detection

Improved Results: As shown below in the figure, the ball in addition to the players were correctly identified with high confidence rates.

Figure 6.2: Success Sample of Object Detection

6.2 Visual Object Tracker

The Visual Tracker module, anchored by an Autoencoder, previously discussed, underwent rigorous prototyping and evaluation to gauge its performance in capturing player movements and generating precise masks. The prototype is designed to demonstrate its effectiveness in tracking individual players throughout a football match and the results from the prototype phase illuminate both successes and areas necessitating further refinement.

6.2.1 Tracking Approach

The tracking process is a collaborative procedure between different computer vision (connected component analysis, geometric transformations) and deep learning (the autoencoder model) techniques to generate the desired result.

The procedure is initiated by providing the model with a cropped image of the player along with its corresponding mask. Subsequently, the model begins masking the player in the designated region of the full match frame. The module is responsible for updating the cropped image presented to the model in each subsequent frame, ensuring the continuous refinement of the tracking process.

The tracking process involves the following key steps:

1. **Image Cropping:** Each frame is initially cropped to focus on the specified window containing the player of interest.

2. **Autoencoder Processing:** The autoencoder model processes the cropped image window to generate a binary mask representing the segmented player.

3. **Connected Components Analysis:** A connected components analysis is applied to identify the player's segmented region within the binary mask.

4. **Bounding Box Calculation:** The bounding box of the player's segmented region is computed, allowing for precise localization.

5. **Adaptive Window Update:** The tracking window is dynamically updated based on the player's movements, ensuring accurate and adaptive tracking.

6.2.2 Results Samples

The Visual Tracker prototype was evaluated on a sequence of frames, and a sample of results are illustrated in the figures. Each sample consists of 2 figures where the first shows the window the autoencoder worked on followed by the connected components analysis used to draw the bbox. The second picture of each sample features the full frame with 2 bbox drawn, the red illustrating the window and the blue one is the final tracked bbox of the player.

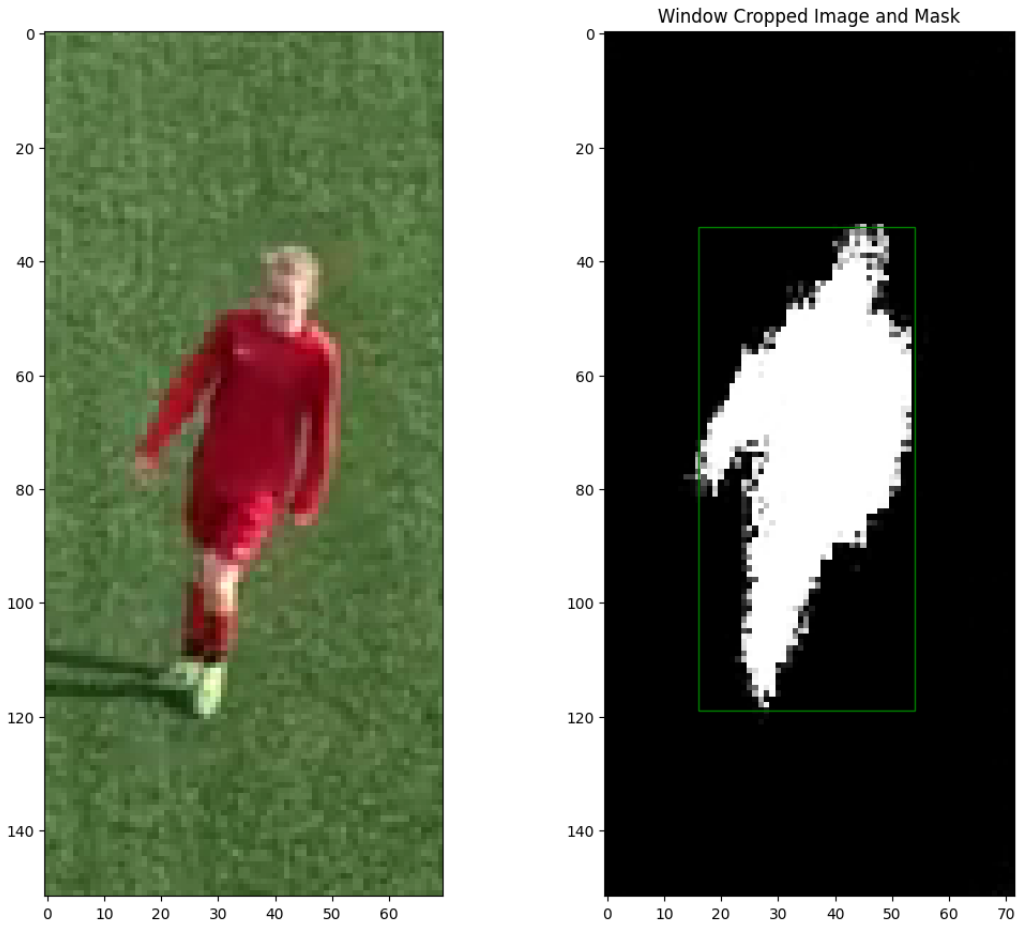


Figure 6.3: Sample 1 of Visual Player Tracking

Figure 6.4: Sample 1 of Visual Player Tracking, Full Frame

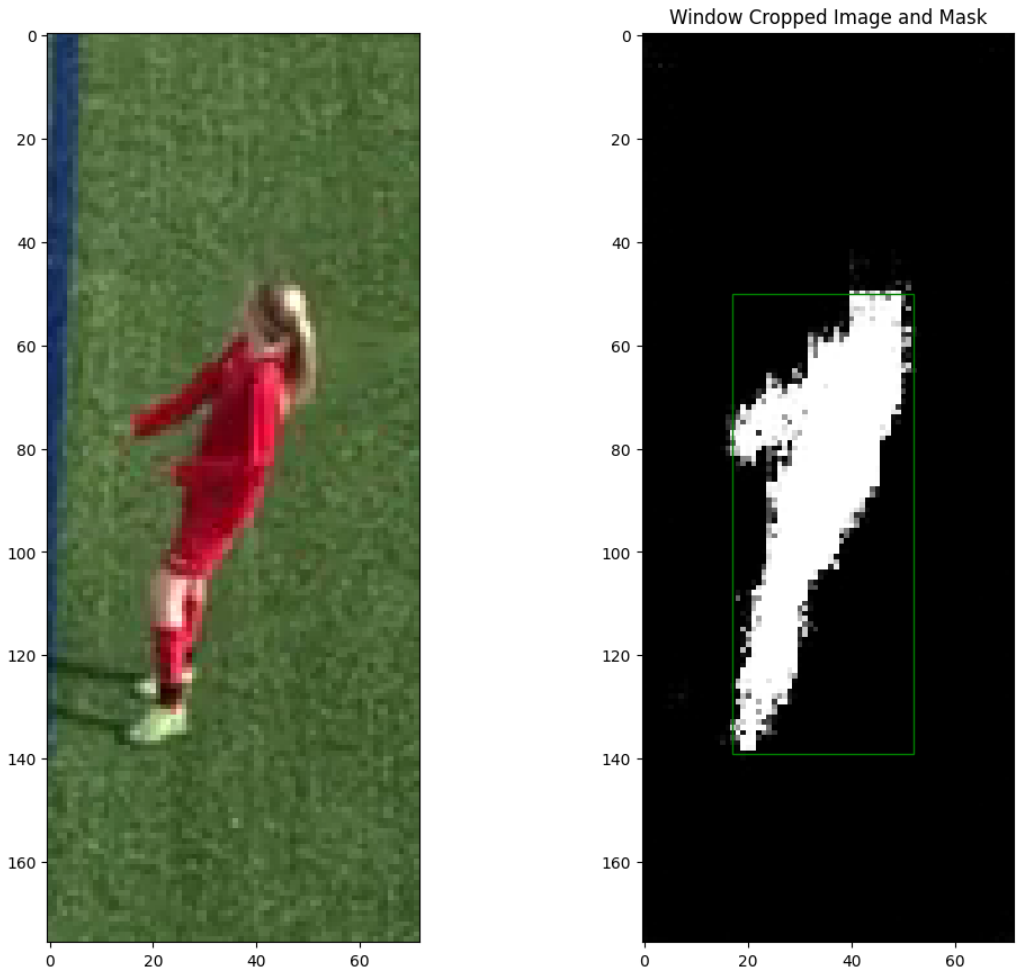


Figure 6.5: Sample 2 of Visual Player Tracking

Figure 6.6: Sample 2 of Visual Player Tracking, Full Frame

While the Visual Tracker prototype demonstrates promising results, challenges persisted in certain scenarios, like having more than a single player in the cropped images or overlapping objects of similar colors leading to misidentifications and sub-optimal tracking accuracy. Further investigation and refinement are imperative to address these challenges and enhance the robustness of the tracking algorithm will undergo in the next steps of our project development.

In the subsequent weeks, the Visual Object Tracker will be integrated into the holistic system architecture, providing insights into its collaborative functioning with other modules and its overall impact on the system's performance.

6.3 Path Prediction

Keeping in mind that this model was in a proof-of-concept phase, so the dataset used to train the model was so small and fixed to one camera's point of view. However, considering these harsh training conditions the model produced fairly reasonable results. The model was trained using approximately 160 samples and reached a dice score around 48

The Dice score equation is defined as:

$$\text{Dice Score} = \frac{2 \times |A \cap B|}{|A| + |B|} \tag{6.1}$$

Where:

$|A \cap B|$ is the number of elements common to both sets A and B

$|A|$ & $|B|$ are the sizes of sets A and B respectively.

6.3.1 Results Samples

This testing was done by using a tracker on the player and checking the output masks with the player's movement afterwards. The model has a little bit of trouble with directions as there is no sense of which direction the player is moving in the input mask, but we are aiming to solve this issue in the future.

Figure 6.7: Sample 1 of Path Prediction

Figure 6.8: Sample 2 of Path Prediction

Figure 6.9: Sample 3 of Path Prediction

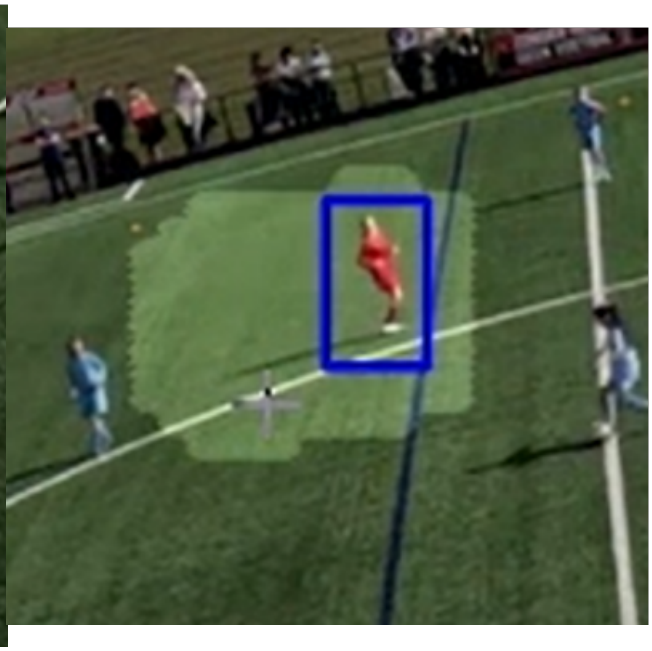


Figure 6.10: Sample 4 of Path Prediction

6.4 Similarity Matching Experiments

The Similarity Module, hinged on Variational Autoencoders (VAEs) as elucidated earlier, has undergone meticulous prototyping and extensive evaluation to assess its efficacy in re-identifying players post-occlusion. The prototype aims to showcase the module's capability in seamlessly matching players before and after occlusion events in the dynamic context of a football match.

The evaluation results from this initial phase shed light on its accomplishments, illustrating successful player re-identification, while also pinpointing aspects that warrant further refinement for enhanced performance. The prototype phase serves as a crucial benchmark in gauging the practical viability and robustness of the Similarity Module in the nuanced landscape of player tracking during occlusion scenarios.

6.4.1 Generating Samples

To validate the successful training of the Variational Autoencoder (VAE) models, separate training sessions were conducted on two distinct players, resulting in the creation of two individual VAE models. Subsequently, a set of samples was drawn from each trained model to assess the efficacy of the generative capabilities. The generated output, obtained through these samples, serves as a quantitative and qualitative measure, affirming the successful learning and encoding of the distinctive features of the respective players. The reported results provide valuable insights into the model's ability to faithfully reproduce meaningful representations of the input data, ultimately verifying the efficacy of the VAE training process for player-specific instances.

Figure 6.11: Player 1 Original Cropped Frames

Figure 6.12: Player 1 Generated Samples

Figure 6.13: Player 2 Original Cropped Frames

Figure 6.14: Player 2 Generated Samples

6.4.2 Similarity Matching Prototype

To substantiate the efficacy of the Similarity Module in the realm of person re-identification, a comprehensive validation process is undertaken employing two distinct Variational Autoencoder (VAE) models. Each model is individually trained on cropped frames corresponding to different players from the same team. The validation procedure is meticulously structured through the following pipeline:

1. **Model Training:**

   Individual Training: Train $model_1$ on a dataset comprising cropped frames specific to player 1. Simultaneously, train $model_2$ on a distinct dataset containing cropped frames pertaining to player 2.

2. **Mean and Variance Storage:**

   $Model_1$ Encoding: Pass a set of cropped frames of $player_1$ through $model_1$, storing the resultant mean and variance representations.

   $Model_2$ Encoding: Pass a separate set of cropped frames of $player_2$ through $model_2$, storing the corresponding mean and variance representations.

3. **Re-identification Attempt:**

Dynamic Assessment: Given a cropped frame depicting either $player_1$ or $player_2$, initiate the re-identification process.

4. **KL Divergence Calculation:**

   $Model_1$ Evaluation: Pass the input frame to $model_1$, computing the minimum Kullback-Leibler (KL) divergence between the derived mean and variance and the stored values associated with $model_1$.

   $Model_2$ Evaluation: Repeat the process for $model_2$, calculating the minimum KL divergence for the input frame against the stored values for $model_2$.

5. **Player Matching:**

   Decision Criteria: Determine the minimum of the computed KL divergences from both models. Match the player in the input frame to the class corresponding to the minimum KL divergence, effectively re-identifying the player post-collision.

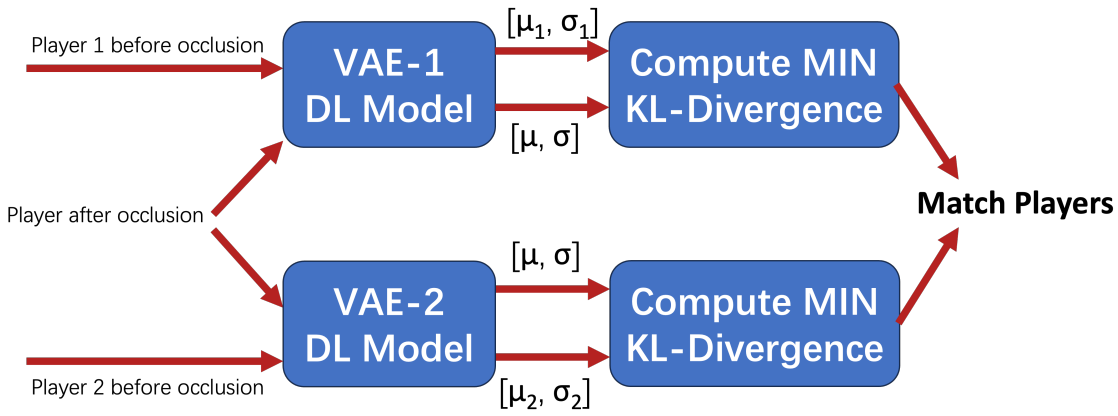


Figure 6.15: Similarity Module Operation

In the evaluation of the re-identification process within the Similarity Module, it is imperative to acknowledge that Kullback-Leibler (KL) divergence is not the sole dissimilarity metric under consideration. Beyond KL divergence, a comprehensive exploration of additional dissimilarity metrics will be undertaken to discern the most effective measure for player matching. Metrics such as Jensen-Shannon (JS) divergence and Wasserstein distance, among others, will be subjected to scrutiny. This multifaceted analysis aims not only to assess the accuracy of the dissimilarity metrics but also to weigh their computational efficiency. The selection of the most suitable dissimilarity metric will be contingent upon a careful balance between time efficiency and accuracy, ensuring that the re-identification process is not only precise but also optimized for real-time applications.

Conclusion

In conclusion, the football analytics system presented here represents a groundbreaking approach to player tracking and analysis in the dynamic realm of football matches. The system's intricate architecture, characterized by a modular design, seamlessly integrates cutting-edge technologies to automate and enhance traditional manual tracking methods. The amalgamation of deep learning advancements with computer vision techniques propels the system beyond the limitations of labor-intensive processes, limited granularity, and occlusion issues.

Our primary goals of developing a modular system, providing accurate player statistics, creating a dynamic user interface, ensuring efficiency in tracking, and facilitating human-agent interaction have guided the design and development of each module. The integration of deep learning advancements and computer vision techniques has allowed us to overcome the limitations of manual tracking, offering a smart toolset for coaches, analysts, and football enthusiasts to gain deeper insights into the beautiful game.

The modular architectural design ensures flexibility, scalability, and efficient communication between components. Each module, whether it be the Object Detection spotlight operator, Visual Tracker persistent conductor, Path Prediction cautious prophet, Similarity dynamic re-identification framework, or Instance Segmentation master of disguise, plays a crucial role in the symphony of player tracking.

As we look forward to the continued development of the Instance Segmentation Module in the upcoming semester, it is imperative to recognize that the success of our football analytics system hinges on the harmonious interplay of all modules. The orchestration of this distributed dance, driven by a deep understanding of football and empowered by cutting-edge technology, is what sets our system apart. In the realm of player tracking, each stand-alone module contributes its unique skills, but their true significance is realized when integrated into the cohesive and intelligent framework we have meticulously crafted.

As we move forward, the system aims to redefine how football analytics is conducted, offering a valuable and smart toolset for coaches, analysts, and football enthusiasts to gain deeper insights into the beautiful game. With future enhancements planned, the system is poised to continue evolving to the ever-changing landscape of football analytics. The journey towards excellence in player tracking and analysis is an ongoing one, and our system stands at the forefront of this transformative endeavor.

Bibliography

[1] Alex A. Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.

[2] Alexandre Alahi, Kshitij Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 961–971. IEEE, 2016.

[3] M. Bai and R. Urtasun. Deep watershed transform for instance segmentation. In *CVPR*, 2017.

[4] David S. Bolme, John R. Beveridge, Bruce A. Draper, and Yui Man Lui. Visual object tracking using adaptive correlation filters. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2544–2550, 2010.

[5] K. Chen and et al. Mask scoring r-cnn. In *arXiv preprint arXiv:1903.00241*, 2019.

[6] X. Chen, M. Treiber, V. Kanagaraj, and H. Li. Social force models for pedestrian traffic–state of the art. *Transportation Reviews*, 38(5):625–653, 2018.

[7] Dai and et al. Instance-aware semantic segmentation via multi-task network cascades. In *CVPR*, 2016.

[8] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6645–6649. IEEE, 2013.

[9] B. Hariharan and et al. Simultaneous detection and segmentation. In *ECCV*, 2015.

[10] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *ICCV*, 2017.

[11] D. Helbing and P. Molnár. Social force model for pedestrian dynamics. *Physical Review E*, 51(5):4282–4286, 1995.

[12] David Held, Sebastian Thrun, and Silvio Savarese. Learning to track at 100 fps with deep regression networks. In *European Conference on Computer Vision*, pages 749–765, 2016.

[13] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[14] H. Ishfaq, A. Hoogi, and D. L. Rubin. TVAE: Triplet-based variational autoencoder using metric learning. 2018.

[15] Rudolf E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960.

[16] A. Kirillov and et al. Instancecut: from edges to instances with multicut. In *CVPR*, 2017.

[17] Colin Lea, Michael D. Flynn, René Vidal, Austin Reiter, and Gregory D. Hager. Temporal convolutional networks: A unified approach to action segmentation. In *European Conference on Computer Vision*, pages 47–54, 2017.

[18] X. Liang and et al. Proposal-free network for instance-level object segmentation. *Arxiv*, 2015.

[19] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *arXiv preprint arXiv:1708.02002*, 2017.

[20] Kevin Noto, Takuya Funatomi, and Michihiko Minoh. Soccernet: A scalable dataset for action spotting in soccer videos. *arXiv preprint arXiv:1902.01169*, 2019.

[21] Frank Pasquale. Toward a fourth law of robotics: Preserving attribution, responsibility, and explainability in an algorithmic society. *Ohio St. LJ*, 78:1243, 2017.

[22] Joseph Redmon and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016.

[23] Joseph Redmon and Ali Farhadi. Yolo v3: Visual and real-time object detection model for smart surveillance systems(3s). In *arXiv preprint arXiv:1804.02767*, 2018.

[24] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[25] Daniel Ridel, Nikhil Deo, Dennis Wolf, and Mohan Trivedi. Scene compliant trajectory forecast with agent-centric spatio-temporal grids. *IEEE Robotics and Automation Letters*, 5(2):2816–2823, 2020.

[26] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1988.

[27] S. Samanta, S. O'Hagan, N. Swainston, T. J. Roberts, and D. B. Kell. VAE-SIM: a novel molecular similarity measure based on a variational autoencoder. *Molecules*, 25(15):3446, 2020.

[28] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5998–6008, 2017.

[30] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2015.

[31] J. Xiang, Z. Huang, X. Jiang, and J. Hou. Similarity learning with deep crf for person re-identification. *Pattern Recognition*, 135:109151, 2023.

[32] Jianglong Ye, Yuntao Chen, Naiyan Wang, and Xiaolong Wang. Online adaptation for implicit object tracking and shape reconstruction in the wild. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022.

[33] Shuai Yi, Hongsheng Li, and Xiaogang Wang. Pedestrian behavior understanding and prediction with deep neural networks. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision–ECCV 2016*, pages 263–279, Cham, 2016. Springer International Publishing.

[34] Chenjuan Yu, Xiaojie Ma, Jie Ren, Hengshuang Zhao, and Shuai Yi. Spatio-temporal graph transformer networks for pedestrian trajectory prediction. In *European Conference on Computer Vision*, pages 507–523. Springer, 2020.

[35] Wenjie Zhang, Hengrong Zhang, and Yang Yang. Learning multi-object tracking and segmentation from event cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9640–9649, 2019.