KTH ROYAL INSTITUTE
OF TECHNOLOGY

Degree Project in Sports Technology

Second Cycle, 30 Credits

Stockholm, Sweden 2024

Comparison of object representations for detected human in football scenes

Ting-Yuan Huang

Author

Ting-Yuan Huang
Master's programme in Sports Technology
School of Engineering Sciences in Chemistry, Biotechnology and Health
KTH Royal Institute of Technology

Supervisors

Håkan Ardö, Haochen Liu
Spiideo

Host Company

Spiideo
Malmö, Sweden

Reviewer

Jonas Willén
KTH Royal Institute of Technology

Swedish Title

Jämförelse av objektrepresentationer för detekterade människor i fotbollsscener

TRITA

CBH-GRU-2024:335

Abstract

Bounding boxes have been widely used in object detection models to represent detected objects. In football scenes where cameras are filming from the side view, problems occur when there are occlusions between the human objects. The aim of this thesis project is to utilize key points from the human body for object representation instead of bounding boxes. Adapted from You Only Look Once (YOLO), which is a real-time object detection model, the regression of bounding boxes was turned into a regression problem of key points in this thesis. Three key points models and one bounding box model were trained on a synthetic dataset for comparison. Non-Maximum Suppression (NMS) during post processing was implemented with key points distances instead of Intersection Over Union (IoU) for the key points model. The performances of the models were evaluated with Precision, Recall, F1 score and Mean Average Precision (mAP). The results indicated that the bounding box model outperforms the key points models while the pelvis and feet points model was identified to perform the best out of the key points models.

Keywords

Object detection, YOLO, Human key points, Computer vision

Abstract

Avgränsningsrutor har använts i stor utsträckning i objektdetekteringsmodeller för att representera detekterade objekt. I fotbollsscener där kameror filmar från sidovyn uppstår problem när det finns överlappning mellan människorna. Syftet med detta examensarbete är att använda nyckelpunkter från människokroppen för att representera objekt istället för avgränsningsrutor. Inspirerad av YOLO, som är en realtidsmodell för objektdetektering, omvandlades regressionen av avgränsningsrutor till ett regressionsproblem med nyckelpunkter i detta arbete. Tre nyckelpunktsmodeller och en avgränsningsrutemodell tränades på en syntetisk datamängd för jämförelse. I postprocesseringen implementerades NMS med nyckelpunktsavstånd istället för IoU för nyckelpunktsmodellen. Modellernas prestanda utvärderades med Precision, Recall, F1-score och mAP. Resultaten visade att avgränsningsrutemodellen presterade bättre än nyckelpunktsmodellerna, medan modellen för bäcken- och fotpunkter identifierades som den bästa bland nyckelpunktsmodellerna.

Nyckelord

Objektdetektering, YOLO, Människans nyckelpunkter, Datorseende

Acknowledgements

First, I would like to express my gratitude towards my supervisors Håkan and Haochen at Spiideo. Their experience and guidance have helped me overcome obstacles throughout the process to complete the thesis. Thank you to the machine learning team and everyone at Spiideo for sharing their knowledge and creating a positive work environment. I am truly grateful to have the opportunity to combine the thesis with my great interest in sports. I would also like to thank the other thesis students at Spiideo. Their companion have made life during the thesis much more enjoyable both inside and outside the office.

I would like to thank Jonas for his advises on how to structure the thesis report and his continuous support during the past two years of the studies. A big thanks also goes to all my classmates in the Sports Technology programme. It has been an amazing journey and the support from them is greatly appreciated.

The computations and data storage in this thesis project was enabled by the Alvis resource provided by Chalmers e-Commons at Chalmers, the Berzelius resource provided by the Knut and Alice Wallenberg Foundation at the National Supercomputer Centre and the National Academic Infrastructure for Supercomputing in Sweden, partially funded by the Swedish Research Council through grant agreement no. 2022-06725.

Stockholm, August 2024
Ting-Yuan Huang

Acronyms

YOLO	You Only Look Once
NMS	Non-Maximum Suppression
IoU	Intersection Over Union
mAP	Mean Average Precision
CNN	Convolutional Neural Network
SSD	Single Shot MultiBox Detector
BEV	Bird's Eye View
FPN	Feature Pyramid Network
CIoU	Complete Intersection Over Union
BCE	Binary Cross Entropy
GPU	Graphical Processing Unit
AP	Average Precision

Contents

Chapter 1

Introduction

1.1 Background

The amount of technology being applied in sports has been increasing rapidly in recent years [1]. In football specifically, tools have been developed to enhance teams and players' performance on the pitch [2]. Examples include GPS sensors, heart rate sensors and video analysis systems. Automatic cameras that film matches and training sessions have gained significant popularity first in professional clubs but now also in youth and amateur football. By understanding the dynamic of the game, the cameras are able to follow the games as if there are people operating the cameras manually. To achieve this, object detection is the first step of the whole process depicted in Figure 1.1.1. After detecting the human objects and locating their positions in the image space, the information gained in image space can then be projected into the world space with known camera parameters. The information in world space including location, running speed, formation of teams etc. is what the players and coaches are interested in [3]. Adding the time component into a sequence of images, tracking could be done to obtain the information mentioned above and also allow the cameras to know where are the areas of interests to focus on. Current technologies often use bounding boxes to represent detected human objects.

1.2 Problem

Problems occur when two or more human objects are overlapping in some frames of a video. This is called occlusion which is a main challenge in the application of object detection in football. For the object being occluded in the image, often the bounding

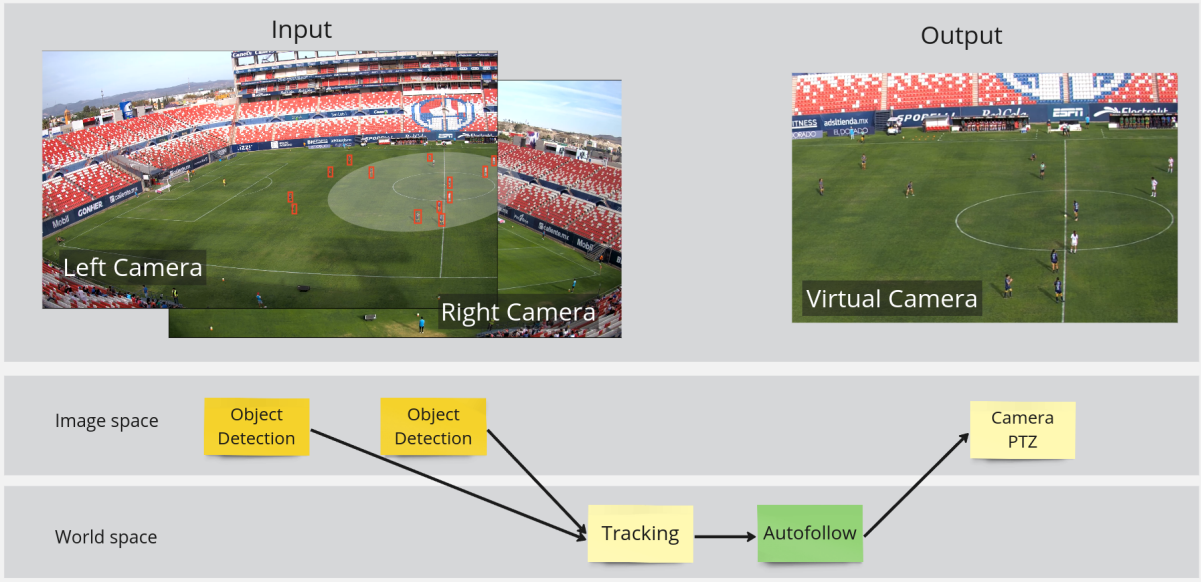


Figure 1.1.1: Overview of auto-follow process.

box will not be able to represent the correct location of the object or the model could not correctly identify the object as shown in Figure 1.2.1. Occlusions can happen in football when there is a corner kick or free kick where players from both teams tend to gather in the same area and are very close to each other.

Another common problem when the camera angle is from the side view is inaccurate localization of detected objects [4]. Since for the bounding box representation the middle point of the bottom edge of the box is taken and projected into world space, this can sometimes be inaccurate when the object is closer to the edges of the image or when the human object is not in a standing pose. Figure 1.2.2 gives an example of an human object detected close to the bottom edge of the image. As it can be seen that the feet of the human is located around the bottom right corner of the box instead of the middle point on the bottom edge of the box.

1.3 Purpose

The purpose of the thesis project is to investigate ways of representing detected human objects by using key points from the human body instead of using bounding boxes. Object detection is the first step for multiple use cases as mentioned in Section 1.1 and the quality of detection results plays a important role for the processes following it such as tracking and projection. Better ways of detected object representation could be beneficial for obtaining more natural views from automatic capture systems and higher quality of

Figure 1.2.1: An example of occlusion.

Figure 1.2.2: Inaccurate localization.

data for analytics. On a wider scale, the utilization of automatic camera capture systems in youth and amateur sports could drastically increase people's engagement in sports, providing positive impacts on health and social aspects.

1.4 Goals

The goal of the thesis project is to utilize human body key points instead of using bounding boxes to represent or obtain information about the detected human objects in frames of football videos. This can be divided into the following sub-goals:

1. Implement anchor key points instead of anchor boxes.
2. Identify the combinations of key points that perform better in detection especially in crowded scenarios.
3. Validate and compare the proposed anchor key points method with a synthetic dataset.

Chapter 2

Theoretical Background

2.1 Object detection

Object detection is an important task in various fields for example autonomous vehicles, medical imaging and industrial automation. In sports, object detection has gained plenty of attention in recent years due to the prevalence of technology introduced into sports [5–7]. It is a computer vision task that tries to identify and localize object in an image or a frame of a video. Compared to image classification, object detection goes one step further specifying the location of the identified objects. Object detection typically involves the following steps:

1. Localization: Localization is the step to determine the objects spatially. A bounding box is often used to represent the detected object. It should enclose the object as tight as possible.

2. Classification: Classification is the step that assigns labels to each detected object. The goal is to distinguish the class or type for all the detected objects.

With the information gathered from object detection for each frame, the next step could be adding the time element for tracking the detected objects. This could involve recognizing different instances of objects within the same class. For example distinguishing multiple players in the same image [8].

2.1.1 Overview of object detectors

It is widely accepted that object detection has gone through the traditional detection period, which is before 2014, and after 2014 is the deep learning-based period [9]. In 2001, P. Viola and M. Jones developed a framework that can detect human faces in real-time [10]. It utilizes sliding windows to go through all possible locations and searches for Haar wavelets which are feature representations in an image. Histogram of Oriented Gradients feature descriptor was proposed in 2005 by N. Dalal and B. Triggs [11]. It is considered as an improvement of the scale invariant feature transform and shape contexts at that time. The two traditional methods above are less common nowadays with most modern approaches utilize deep learning-based methods.

In the deep learning-based period, object detectors can be categorized into two groups, which are two-stage detectors and one-stage detectors. Two-stage detectors first propose regions of interest, then classify and refine these proposals. An example of a two-stage detector is the Faster R-CNN [12]. On the other hand, one-stage detectors perform detection and classification in a single step. They directly predict the locations and class probabilities for all potential candidates. In general, one-stage detectors are faster and works in real-time applications while two-stage detectors tend to achieve higher accuracy. The following sections introduce some other detectors related to the thesis project.

2.2 CNN-based object detectors

Convolutional Neural Network (CNN) is a type of deep learning algorithm inspired by the human visual cortex specifically designed for processing visual data and obtaining visual information. CNNs usually consist of convolutional layers, pooling layers and fully connected layers. The following is a short introduction of each layer:

1. Convolutional layers: Convolutional layers perform convolution on the input image to extract features such as edges, corners and patterns. The layers produce feature maps which are passed on the the pooling layers.

2. Pooling layers: The main function of pooling layers is to increase the robustness of the network against variations and decrease computational complexity while still keeping the most important information of the feature maps. Common methods include max pooling and average pooling.

3. Fully connected layers: Fully connected layers typically exist closer to the end

of the network. These layers connect every neuron from the previous layer to the subsequent layer, usually performing classification and generating the final outputs.

CNNs learn to adjust its weights and biases to make the predicted outputs as similar to the ground truth labels as possible through the process of backpropagation. The training process involves forwarding the input data, computing the loss and optimizing the parameters by calculating the gradient of the loss function with algorithms like Stochastic Gradient Descent.

2.2.1 Single Shot MultiBox Detector

Single Shot MultiBox Detector (SSD) is a one-stage detector that was presented by W. Liu et al. in 2015 [13]. SSD discretizes the output space into a set of anchor boxes with various aspect ratios and scales. During prediction, the network produces scores for each object category and adjustments to the boxes for each anchor box. Unlike two-stage detectors, the method includes all computations in a single network. The method also introduces multi-reference and multi-resolution detection techniques, achieving better detection accuracy similar to two-stage detectors while performing at a faster speed. Figure 2.2.1 illustrates the framework of SSD.

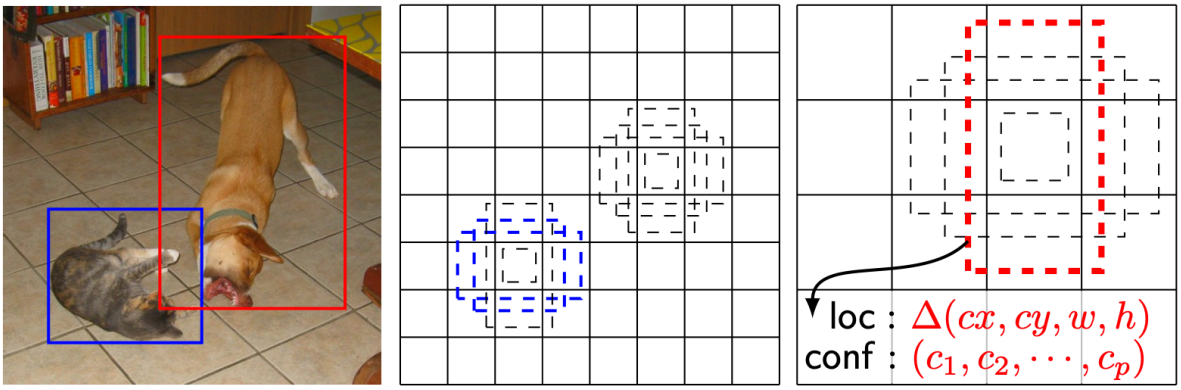


Figure 2.2.1: The framework of SSD. The image to the left is the input image with ground truth boxes. The images in the middle and to the right are feature maps with different aspect ratios at different scales. Note that only boxes in certain locations are showed [13].

2.2.2 CenterNet

Introduced by X. Zhou et al. [14], CenterNet takes a different approach by considering an object as a point (the center point). The method uses key point estimation to find

the center points and regresses to the other attributes such as size, orientation and pose. CenterNet first finds the heatmap for each category where each peak corresponds to the center of an object of that class. After detecting the centers of objects, CenterNet predicts the offset and size of the bounding box directly without the need for IoU-based NMS. Compared to other key point-based object detection methods such as CornerNet [15] and ExtremeNet [16], CenterNet eliminates the step of grouping or post-processing key points after detection. This allows CenterNet to be able to run in real-time while getting competitive results as two-stage detectors. The modelling of an object in CenterNet is illustrated in Figure 2.2.2. Notice that there are no anchors involved in CenterNet.

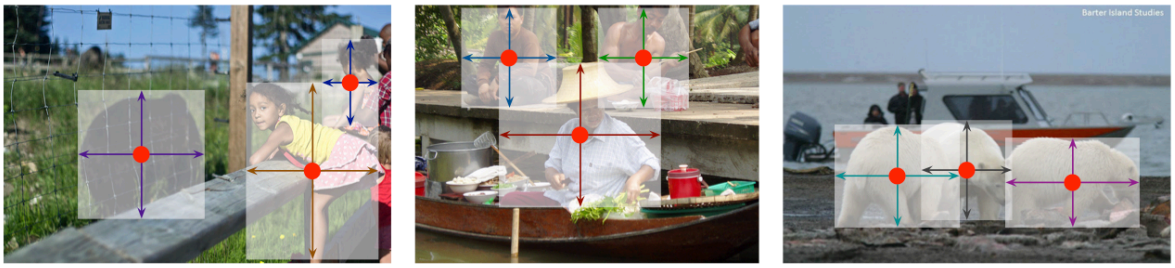


Figure 2.2.2: The objects are modelled as the center point of their bounding boxes. All the properties are inferred from the center point [14].

2.2.3 You Only Look Once

YOLO is a real-time object detector first presented by J. Redmon et al. [17] that predicts bounding boxes and class probabilities in one forward pass through the network. By dividing the input image into grid cells, each grid cell is supposed to detect the object whose center lies in it. Figure 2.2.3 illustrates the working process of the original system. The model treats detection as a regression problem. Confidence thresholding and NMS is performed during post-processing to remove excessive predictions. After the release of the first version, the model has drew wide attention due to its speed, which is the biggest advantage of YOLO compared to other object detectors at the same time. YOLOv3 [18] included detection at three different scales by incorporating Feature Pyramid Network (FPN) [19], which will be introduced in Section 2.4. This improved the detection performance especially for smaller objects. YOLOv5 [20] implemented advanced deep learning techniques such as mosaic data augmentation and further improved the model performance. YOLOv7 [21, 22] introduced more optimizations for example by modifying the network structure and achieved even better speed and accuracy.

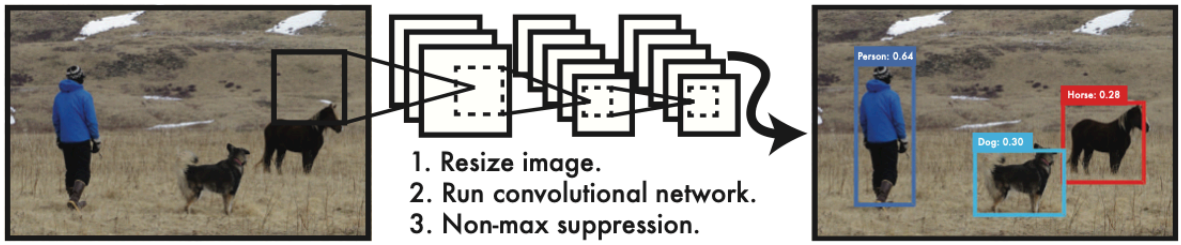


Figure 2.2.3: Overview of the YOLO system [17].

2.2.4 Bird's Eye View-based methods

The Bird's Eye View (BEV) method in object detection refers to performing detection from an top-down perspective by projecting all views onto the ground plane. It is particularly useful for applications that need spatial information between the objects. The main concept involved is view transformation. In [23], image space coordinates are converted into world space through light ray projection and the known intrinsic and extrinsic camera parameters. A two-stage object decoder with view combination techniques and a depth estimation module was introduced in [24]. The framework proposed in [25] aimed to address the problem of inaccurate depth estimation between cars and the ground. These methods has mostly been focusing on the use case of autonomous driving in which the spacial relationship information for the objects around the car is important. By further combining multi-view angles into BEV, detection and tracking performance was improved as demonstrated in [26]. Although view transformation can be complex, aggregating multi-views and projecting all of them to the ground place shows great potential in overcoming the problem of occlusion and missed detection.

2.3 Object representation

Object representations in object detectors refers to how detected objects are being represented in the detector framework. It is a crucial part for identifying and localizing objects since these representations store essential information of the objects such as size and position. The following sections introduce and compare three different ways of object representation.

2.3.1 Bounding boxes

Bounding boxes are rectangular frames placed around objects to describe the spatial context of the detected objects. They are usually defined by the coordinates of the top-left and bottom-right corners of the boxes. To provide accurate information about the object's size and location, a bounding box should be the smallest possible rectangle enclosing the whole object with the edges of the box aligned with the image axes. Except for 2D bounding boxes as shown in Figure 2.3.1, there are also 3D bounding boxes as illustrated in Figure 2.3.2 and rotated 2D bounding boxes [27]. With its simplicity, bounding boxes are often utilized in real-time applications or devices with limited computational power.

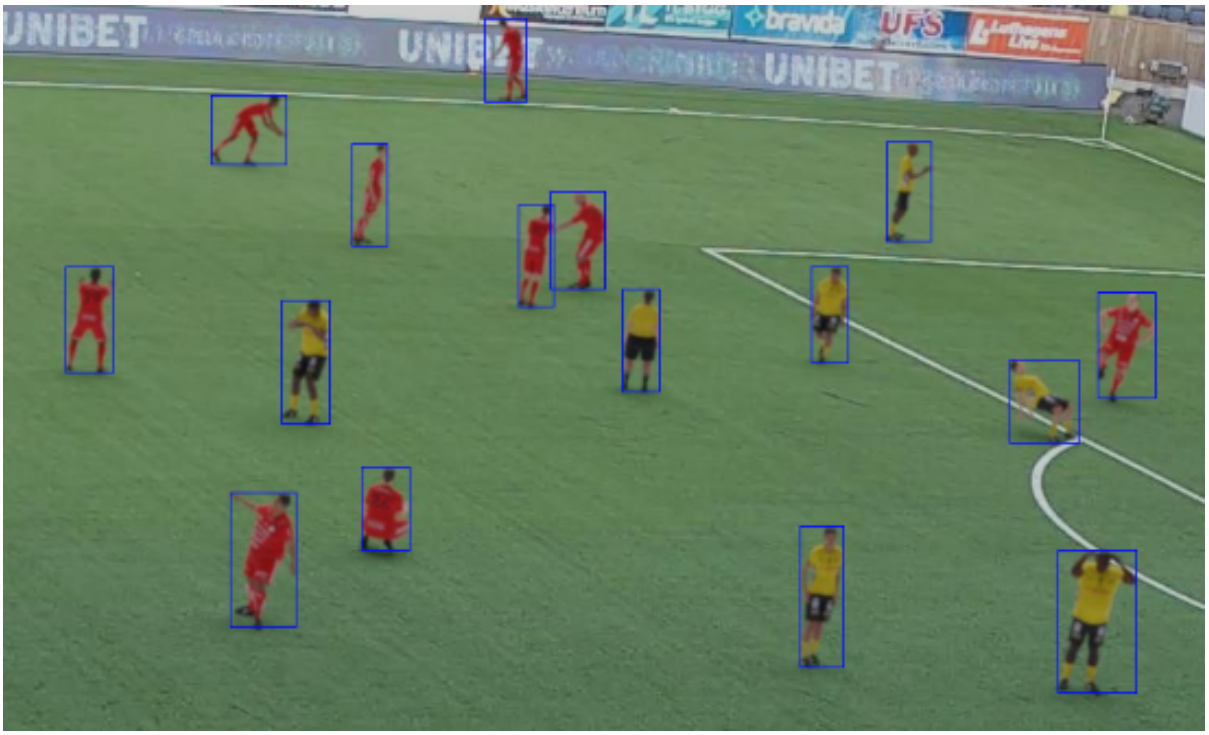


Figure 2.3.1: Examples of 2D bounding boxes.

2.3.2 Key points representation

Key points representation refers to identifying specific points on objects. They help to providing more accurate localization of objects. The key points being identified can vary based on the detected objects and application. Common examples include corners [15], edge points etc. In [28], the authors presented a finer representation of objects as a set of sample points for both localization and recognition. A joint prediction scheme modified from YOLO was proposed in [29] to tackle the challenge of detection in crowded scenes.

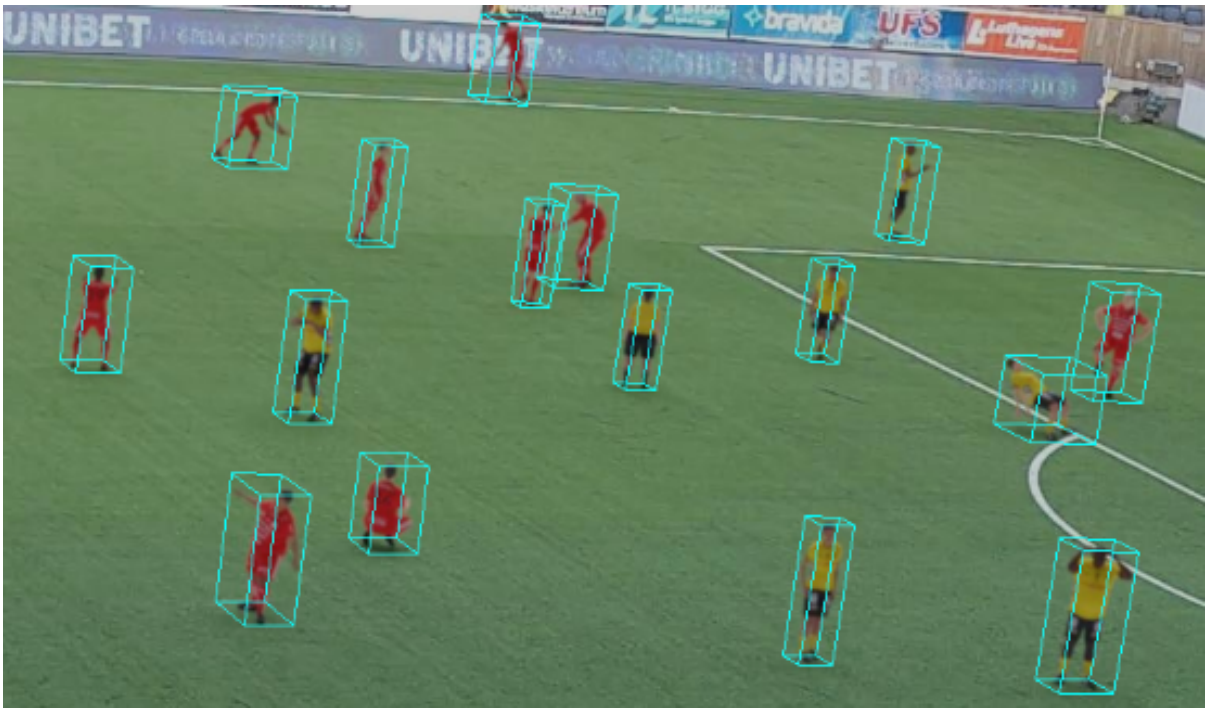


Figure 2.3.2: Examples of 3D bounding boxes.

On a human body, key points can be the locations of elbows, knees, head, fingers, toes, or any other anatomical location. These human body points are used in the task of human pose estimation which the goal is to detect and estimate the position and orientation of a human body. In Figure 2.3.3, the head, feet and wrist points are demonstrated.

Figure 2.3.3: Examples of key points representation.

2.3.3 Instance segmentation

Instance segmentation is another technique to represent detected objects. It takes one step further compared to the other two methods mentioned in the above sections by segmenting each object instance at the pixel level, where each pixel is assigned to a specific object instance. In instance segmentation, pixel-wise masks are generated which creates more detailed boundaries and shapes of the objects. Figure 2.3.4 is an example of segmented human objects. Compared to bounding boxes and key points, instance segmentation requires more computational resources. It is suitable for cases where separating instances of the same object class or precise information such as object boundaries are needed.

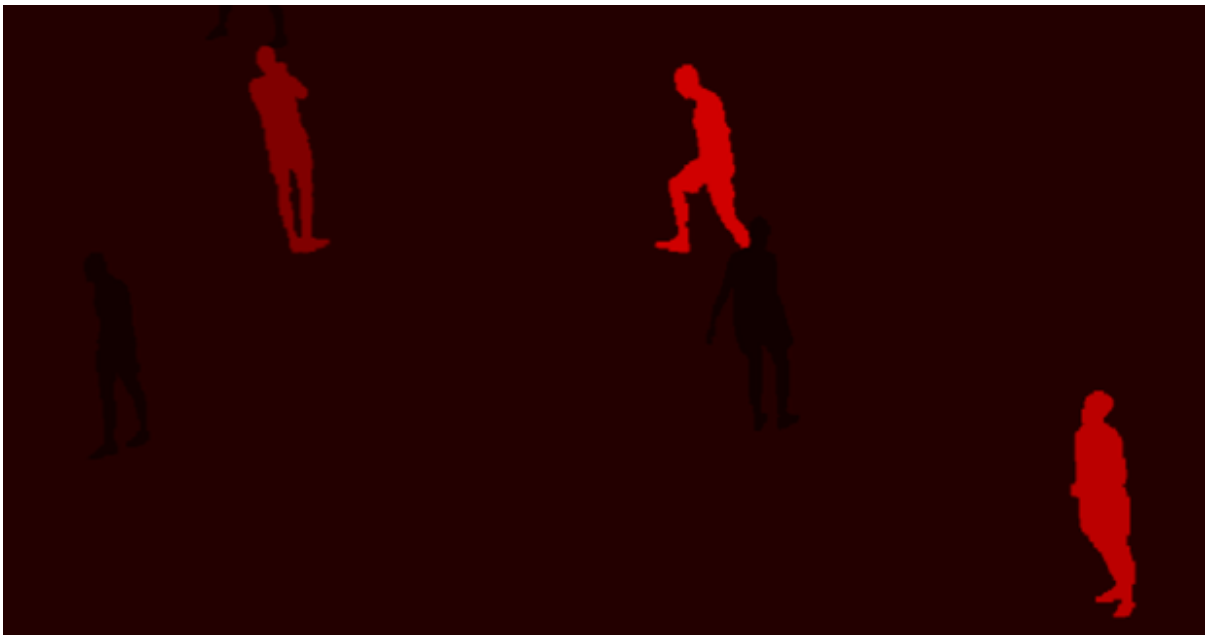


Figure 2.3.4: Examples of instance segmentation.

2.4 Feature Pyramid Networks

FPN is a type of neural network architecture that improves the ability of models for detecting objects at different scales. It was first introduced by T.-Y. Lin et al. in 2017 [19]. Figure 2.4.1 shows the structure of a FPN. It adopts a deep convolutional network [30] and builds a feature pyramid where predictions are made independently on each level.

The pyramid first goes through a bottom-up pathway which is the feed-forward computation of the network. Each stage in the bottom-up pathway computes a feature

map at several scales with a scaling step of two. As a result, the higher pyramid levels have spatially coarser but semantically stronger feature maps.

The top-down pathway and lateral connections upsample the higher level feature map and merge it with the feature map with the sample size from the bottom-up pathway. By doing this, the high-level features are combined with higher resolution features to generate high-resolution and sementically strong features.

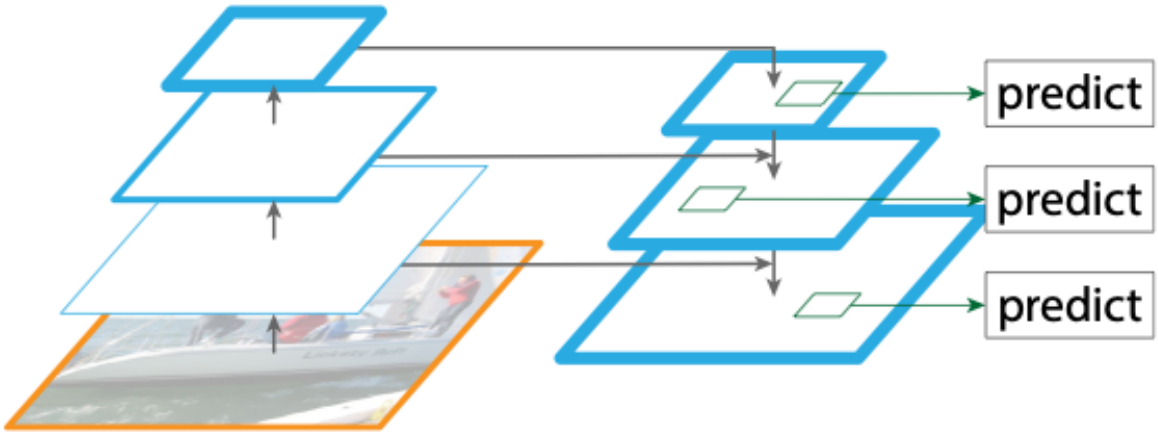


Figure 2.4.1: The architecture of FPN [19].

Chapter 3

Methodology

An overview of the methodology and work flow of this thesis project is presented in Figure 3.0.1. The chapter starts with an introduction of the datasets being used for training and testing in this thesis. Following this, the implementation of anchor key points and modification of the loss function in YOLOv7 is outlined. After introducing the setup for the training of the models, the post-processing part with NMS in both image space and world space is explained. Lastly, the chapter concludes with a description of the evaluation metrics utilized in this thesis project.

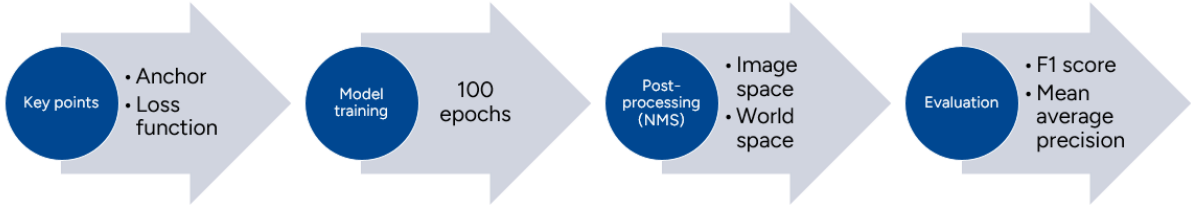


Figure 3.0.1: Overview of the research methodology and process.

3.1 Synthetic datasets

Synthetic data refers to data that is artificially generated rather than collected from the real world. In this thesis project, two synthetic datasets were made ready by the host company Spiideo. It is time-consuming and hard to get a large amount of well-annotated real world data. Instead of manually annotating real data, synthetic data could be generated and varied for different scenarios in much less time. Its flexibility and scalability provides advantages and is hence chosen for the thesis project. One dataset

is named SoccerScene in which each scene consists of Allsvenskan[1] backgrounds with two 11-player teams, three referees and four bystanders. An example scene is shown in Figure 3.1.1 and Figure 3.1.2. The other is the SoccerCrowd dataset. Each scene in SoccerCrowd is rendered with Allsvenskan backgrounds with 60 bystanders and a total of 60 players and referees. Figure 3.1.3 and Figure 3.1.4 illustrate a scene from the SoccerCrowd dataset. In both datasets, there are two images per scene with one aiming at the left side of the field and the other to the right. Those human objects visible in both images are counted twice. Table 3.1.1 provides an overview of the two datasets.

Figure 3.1.1: Left view of a scene from the SoccerScene dataset.

Table 3.1.1: Summary of the datasets.

Dataset	**Images**	**Annotations**
SoccerScene	80000	1302833
SoccerCrowd	4280	289607

For each image, there is a corresponding file that saves the information about the annotations. The list below outlines parts of the information that is used in this thesis project:

- Bounding box coordinates in image space.
- Bounding box coordinates in world space.

[1]Allsvenskan is a professional league and the top division in the Swedish football league system.

Figure 3.1.2: Right view of a scene from the SoccerScene dataset.

Figure 3.1.3: Left view of a scene from the SoccerCrowd dataset.

- Key points coordinates in image space.
- Key points coordinates in world space.

The key points mentioned above include root, pelvis, hip, knee, foot, head, elbow, shoulder, etc. Notice that the origin in image space is at the top left corner of the image, with positive x-direction pointing towards right and positive y-direction pointing

Figure 3.1.4: Right view of a scene from the SoccerCrowd dataset.

down in the image. In world space, the origin is at the middle point of the halfway line on the field. With respect to the camera's perspective, the positive x-direction is pointing away from the camera along the halfway line. The positive y-direction is pointing towards the left from the camera's view. The units for the data are number of pixels and meters for image space and world space, respectively.

3.1.1 Dataset separation

The SoccerScene dataset as introduced in Section 3.1 was separated into 80% training data, 10% validation data and 10% testing data. The SoccerCrowd dataset was not separated and only used in testing.

3.2 Key points model implementation

The object detection model used in this thesis project is YOLOv7. The following sections explain how the key points models are constructed and implemented based on the original YOLOv7 bounding box model. In total, three different key points model were created. The list below consists of the combination of key points for each model. Figures 3.2.1 to 3.2.3 visualize the target key points for each model.

- Pelvis and feet points.

- Pelvis and head points.
- Head and feet points.

Figure 3.2.1: Pelvis and feet points targets.

Figure 3.2.2: Pelvis and head points targets.

3.2.1 Targets matching

YOLOv7 is an anchor-based model. During training, the basic idea of the bounding box model is to first create many potential bounding boxes, select the best possible options and slightly moving and resizing them to obtain the best possible fit to match the target objects. The targets objects are the ground truth labels that exist in the datasets. One way to achieve this is to place the same set of anchor boxes at each anchor grid point.

Figure 3.2.3: Head and feet points targets.

The size of the boxes are pre-defined. Notice that for each detection layer, the aspect ratios of the anchor boxes are not the same. It is worth mentioning that there are three anchor boxes in each detection layer. Compared to generating the boxes from scratch, the anchor grid points with reoccurring boxes made it easier for the model to learn how to relocate and reshape these potential bounding boxes.

Similar to the case of bounding box, a set of anchor key points are created by taking the corresponding points from the anchor boxes. For a human object, the middle point of the left foot and the right root is selected to represent the feet location. The following list shows the key points of the human body with their assumed location on the box.

- Pelvis corresponds to the middle point of the box.
- Feet corresponds to the middle point on the bottom edge of the box.
- Head corresponds to the middle point on the top edge of the box.

The first step in building the targets is to match the targets with the anchors. In the bounding box case for each anchor box, only targets within four times larger or 0.25 times smaller than the the anchor box are considered. That is, the height of the target cannot be larger than four times or smaller than 0.25 times the height of the anchor box. The same applies to the width as well. Figure 3.2.4 demonstrates an example for this criterion.

For the key points models, the same criterion is implemented. Instead of checking the height and width, only the distance between the key points for example pelvis and feet is

looked at. In other words, at the grid where the target is located, as long as the distance between the target key points is within the range of 0.25 to four times the distance of the anchor key points the target could be matched to them.

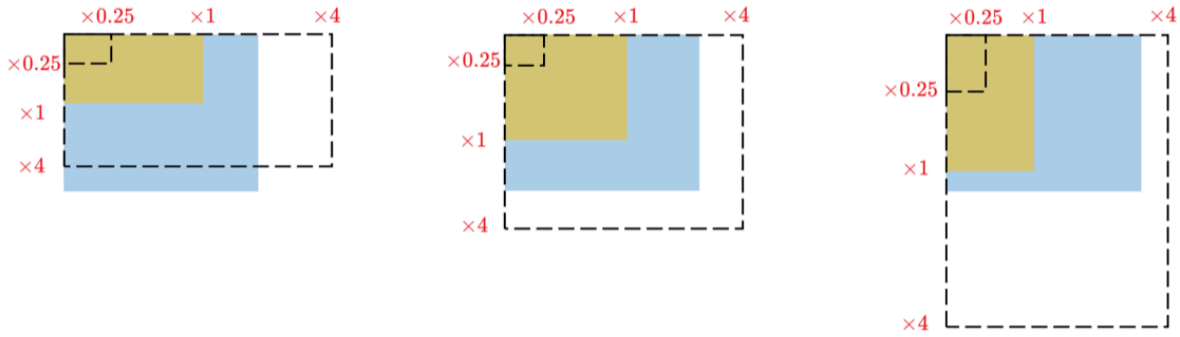


Figure 3.2.4: The blue box is a target and the yellow boxes are a set of three different anchor boxes. The anchor box to the left would not be matched to the target [31].

During training of machine learning models, having enough positive targets is crucial to avoid class imbalance and improve model generalization. For object detection models, the negative class is the background in images where there are no positive targets located. It is common that most part of the images doesn't consist of positive targets. If the ratio of positive class to negative class is too low, the model might learn to only produce negative predictions which is not ideal.

To increase the number of positive samples, the original bounding box model takes the same anchor boxes located at the adjacent grids as positive samples based on the location of the center of the target. For example, if the center of a target is in the top left part of a grid, the same anchor boxes in the grid above and to the left of that grid would also be considered as positive samples as shown in Figure 3.2.5.

As for the key points models, for each matched anchor key points, the amount of positive samples are increased by matching with the two closest same anchor key points in the other grids. Usually they are located in two of the adjacent grids to the grid where the target is at. Information including the grid index where the target is at, which anchor the target is matched to and the distance from the target key points to the matched anchor key points is saved and will be used later for calculating the loss.

3.2.2 Loss function

The loss function in YOLOv7 consists of regression loss, objectness loss and classification loss. The predictions from each detection layer are mapped before calculating the loss.

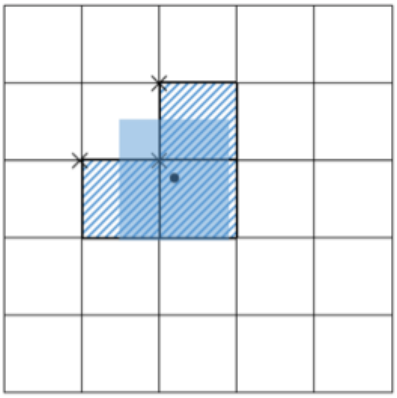


Figure 3.2.5: More positive samples identified depending on the location of the center point on the grid. The black dot represents a center point of a target [31].

Equations 3.1 to 3.4 show the operations applied to the original output in the bounding box case with σ denoting the sigmoid function. In Equations 3.1 and 3.2, x and y represent the x-coordinate and y-coordinate of the center of the box. The mapped coordinates $(x_{\text{mapped}}, y_{\text{mapped}})$ are obtained by

$$x_{\text{mapped}} = 2 \cdot \sigma(x_{\text{raw}}) - 0.5, \tag{3.1}$$

$$y_{\text{mapped}} = 2 \cdot \sigma(y_{\text{raw}}) - 0.5, \tag{3.2}$$

where $(x_{\text{raw}}, y_{\text{raw}})$ represent the raw coordinates. On the other hand, w and h in Equations 3.3 and 3.4 is the width and height of the box. The mapped width and height $(w_{\text{mapped}}, h_{\text{mapped}})$ are obtained from the raw width and height $(w_{\text{raw}}, h_{\text{raw}})$ using

$$w_{\text{mapped}} = (2 \cdot \sigma(w_{\text{raw}}))^2 \cdot w_{\text{anchor}}, \tag{3.3}$$

$$h_{\text{mapped}} = (2 \cdot \sigma(h_{\text{raw}}))^2 \cdot h_{\text{anchor}}. \tag{3.4}$$

Note that during loss calculation grid coordinate is used. For the key points cases, only Equations 3.1 and 3.2 are used to map the predictions as neither the width nor height is involved.

After mapping, the Complete Intersection Over Union (CIoU) [32] between all targets and

their candidate boxes are calculated. The regression loss for bounding boxes is defined as the mean of $(1 - \text{CIoU})$ between all targets and their predicted candidate boxes. For the key points models instead of calculating CIoU, the distances between all target key points and their candidate key points is the metric being looked at. With the target coordinates $(x_{\text{target}}, y_{\text{target}})$ and the mapped prediction coordinates $(x_{\text{mapped}}, y_{\text{mapped}})$, the distance D_{kp} can be calculated by

$$D_{\text{kp1}} = \sqrt{(x_{\text{mapped1}} - x_{\text{target1}})^2 + (y_{\text{mapped1}} - y_{\text{target1}})^2},$$

$$D_{\text{kp2}} = \sqrt{(x_{\text{mapped2}} - x_{\text{target2}})^2 + (y_{\text{mapped2}} - y_{\text{target2}})^2}.$$

For each prediction subset, the regression loss for key points is hence defined as

$$L_{\text{reg}} = \frac{1}{n} \sum D_{\text{kp1}} + \frac{1}{n} \sum D_{\text{kp2}}, \tag{3.5}$$

which is the sum of the mean of distance for each key point as given in Equation 3.5, where n is the number of predictions in the subset. In Equation 3.5 there is always two key points distances since all the key points models in the thesis project are implemented with a combination of two key points. Notice the difference between the two regression losses since the loss should be larger when the distance is larger for key points, and for bounding boxes the loss is small when the CIoU is large.

Objectness is a measure of how confident the model is about the existence of an object in each prediction. The ground truth objectness is calculated by the CIoU mentioned in the previous paragraph for bounding boxes. Binary Cross Entropy (BCE) loss [33] between predicted confidence value and the CIoU is used to obtain the objectness loss. For the key points models, the ground truth objectness T_{obj} is calculated as

$$T_{\text{obj}} = \begin{cases} 1, & \text{if } \frac{1}{D_{\text{kp1}}} + \frac{1}{D_{\text{kp2}}} \geq 1 \\ \frac{1}{D_{\text{kp1}}} + \frac{1}{D_{\text{kp2}}}, & \text{otherwise.} \end{cases} \tag{3.6}$$

Same as for bounding boxes, BCE loss is used for getting the objectness loss. The last element in the loss function is the classification loss. It also utilizes the BCE loss to get the loss between the predicted class probabilities and the ground truth class labels.

For the key points models, no modification was implemented. The classification loss is only calculated when there are multiple classes in the targets. In this thesis project, there is only one human target class in the datasets making the classification loss to always be zero. After obtaining all the components by summing up the values from each prediction subset, each component is multiplied by their predefined contribution weight. The weighted loss is then summed up and multiplied by the batch size, which is the final loss value.

3.3 Model training

The software environment used in the thesis project is PyTorch. All models were trained for 100 epochs on a single Graphical Processing Unit (GPU). The GPUs are located on Berzelius which is the main cluster at the National Supercomputer Centre and the Alvis cluster which is a resource dedicated to artificial intelligence and machine learning research.

3.3.1 Hyperparameters and weights

The models were trained from scratch which means that no pre-trained weights were used. No data augmentation technique was implemented during the training processes. Out of all the hyperparameters, everything including learning rate, weights for the loss components etc. was kept the same from the original YOLOv7 [21] except for the loss was computed without Optimal Transport Assignment [34] for faster training.

3.4 Post-processing

The post-processing steps in YOLO refine the network predictions and produce the final detections. These steps include filtering the predictions with a confidence threshold, NMS to remove similar but excessive predictions and assigning the class labels. The following sections introduce the original NMS method in the bounding boxes case which utilizes IoU and the implementation of NMS with key points distances in both image space and world space. Usually NMS is being done in the image space. Since projection into world space is possible with the known camera metrics and the potential of minimizing the effect of occlusion after projection, NMS in world space is also implemented in the thesis project.

3.4.1 Non-Maximum Suppression in image space

The NMS algorithm for bounding boxes can be summarized as below:

1. Sort the predictions with descending confidence values.
2. Save the prediction with the highest confidence value as it is kept and remove it from the original prediction list.
3. Calculate the IoU between the saved prediction and all the other candidate boxes.
4. Discard the predictions that have IoU values larger than the threshold.
5. Repeat the above steps until the original prediction list is empty.

Based on the above, the NMS algorithm are implemented with key points distances instead of IoU. The following shows the steps of the algorithm:

1. Sort the predictions with descending confidence values.
2. Save the prediction with the highest confidence value as it is kept and remove it from the original prediction list.
3. Calculate the distances between the saved prediction and all the other candidate key points.
4. Normalize the key points distances with the saved prediction object size. The x-coordinates $(x_{\text{pelvis}}, x_{\text{feet}}, x_{\text{head}})$ and y-coordinates $(y_{\text{pelvis}}, y_{\text{feet}}, y_{\text{head}})$ are taken to approximate the object size denoted as S by

$$S = 2 \cdot \sqrt{(x_{\text{pelvis}} - x_{\text{feet}})^2 + (y_{\text{pelvis}} - y_{\text{feet}})^2}, \tag{3.7}$$

$$S = 2 \cdot \sqrt{(x_{\text{pelvis}} - x_{\text{head}})^2 + (y_{\text{pelvis}} - y_{\text{head}})^2}, \tag{3.8}$$

$$S = \sqrt{(x_{\text{head}} - x_{\text{feet}})^2 + (y_{\text{head}} - y_{\text{feet}})^2}, \tag{3.9}$$

   for the pelvis feet, pelvis head and head feet models, respectively. Notice that the distance between head and feet as shown in Equation 3.9 is directly used to approximate the object size while the other two are multiplied by a factor of two.

5. Discard the predictions that have both normalized key points distances smaller than the threshold.

6. Repeat the above steps until the original prediction list is empty.

It is assumed that the object size is two times the distance between the pelvis point and feet point, two times the distance between the pelvis point and head point and the distance between head point and feet point.

Intersection Over Union threshold

The IoU threshold was set to 0.65 in YOLOv7 and was kept the same in this thesis project.

Distance threshold

For the normalized distance threshold, multiple experiments were done on the pelvis and feet points model to try out different threshold values. The distances are in units of pixels and the normalized distance is without units. According to Table 3.4.1, 0.25 was selected as the normalized distance threshold as it produced the highest F1 score and mAP value.

Table 3.4.1: Comparison of different normalized distance threshold.

Threshold	**F1 score**	**mAP@.1:1.0**
0.1	0.75	0.645
0.15	0.76	0.643
0.2	0.79	0.673
0.25	0.81	0.686
0.3	0.81	0.676
0.35	0.8	0.666
0.4	0.8	0.65
0.45	0.8	0.642
0.5	0.79	0.633
0.55	0.79	0.628

3.4.2 Non-Maximum Suppression in world space

For each scene in the datasets, there is information about the camera matrix, parameters of the lens etc. This made it possible to project points from the image space into the world space. The idea is to project the feet points from the key points model and the

middle point of the bottom edge of the boxes from the bounding box model into the ground plane on the pitch. After projection, NMS is performed in world space according to the distances between the projected points. Since only one point is projected from each prediction, the distances are not normalized by the size of the object when applying NMS. Table 3.4.2 shows the results using the pelvis and feet points model. A distance threshold of 0.9 meter was chosen since it had the highest mAP value despite having the same F1 score with other thresholds.

Table 3.4.2: Comparison of different distance threshold in world space.

Threshold (m)	**F1 score**	**mAP@.1:1.0**
0.3	0.42	0.282
0.4	0.43	0.288
0.5	0.44	0.294
0.6	0.45	0.298
0.7	0.46	0.299
0.8	0.46	0.303
0.9	0.46	0.304
1.0	0.46	0.303
1.5	0.45	0.284

3.5 Evaluation metrics

In order to evaluate and compare the performance of the three key points models and one bounding box model, the common machine learning model evaluation metrics including precision, recall, F1 score and mAP are used in this thesis project. The metrics for bounding boxes are based on the IoU values. Similar to the modification done in Section 3.4.1 and 3.4.2, the metrics are implemented with key points distances allow comparisons with the key points models. The following sections give a more detailed introduction to the metrics.

3.5.1 Detection performance

Precision is defined as the ratio between the number of true positive predictions and the number of total predictions as given in Equation 3.10. TP denotes the number of true positives and FP represents the number of false positives. A high precision rate indicates that out of the positive predictions, the ratio of correct predictions is high as well.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3.10}$$

Recall is the ratio between the amount of true positives and the amount of actual positive instances. It is a metric that implies the model's ability to identify the actual positive instances. Equation 3.11 shows the definition of recall where FN denotes the number of false negative predictions.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3.11}$$

F1 score is the harmonic mean of precision and recall. It is defined as in Equation 3.12. It combines both precision and recall into a single number with a higher F1 score indicates a better balance between precision and recall. In the use case of player detection on football pitches in this project, a balance of precision and recall is emphasized hence F1 score is the main metric used for comparing the different models.

$$\text{F1 score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.12}$$

For the bounding boxes, an IoU value larger than 0.5 is defined as a positive detection. In image space, it is considered as a positive detection if both normalized distances between the predicted and target key points are smaller than 0.5. The key points distances are normalized by the average size of the prediction and the target key points. In world space, a positive detection is defined when the key points distance is smaller than 0.5 meter. It is worth mentioning that one target can only be matched to one prediction. If more than one prediction meets the above definitions to the same target, the one with the highest IoU value or the smallest distance is considered. Similarly, a prediction is limited to be matched with one target. The same criteria applies when one prediction could be matched to multiple targets.

3.5.2 Average Precision

Another common metric for evaluating object detection models is Average Precision (AP). It calculates the area under the precision-recall curve providing a numerical metric to compare different object detectors. When looking at a precision-recall curve, it usually starts with high precision values and decreases with increasing recall values. A good

object detector is considered when the precision stays high as the recall increases, hence a higher AP value represents a better detection performance. The the average AP across all object classes is the mAP.

In the original YOLOv7 bounding boxes case, mAP is calculated from a IoU value larger than 0.5 to 0.95 with an increment of 0.05 for considering a positive detection. For the key points models, a normalized distance in the image space from 0.1 to 1 with an increment of 0.1 is defined for mAP calculation. In the world space, a positive detection is defined from 0.1 to 1 meter with an increment of 0.1 for calculating mAP.

Chapter 4

Results

4.1 Training of models

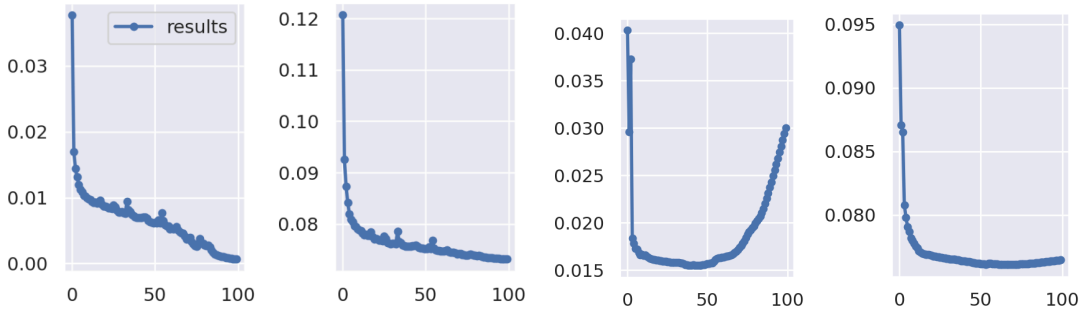
This section presents the loss values during the training processes. As introduced in Section 3.2.2, the loss function is composed of the regression loss, the objectness loss and the classification loss. There was no classification loss since the dataset only consisted of one target object class.

4.1.1 Key points models

Figures 4.1.1 to 4.1.3 show the history of objectness loss and regression loss during training for the three key points models. The x-axis is the number of epochs done during training and the y-axis is the loss value without units. Figure 4.1.1a and 4.1.1b are the losses from the training set while Figure 4.1.1c and 4.1.1d are obtained from the validation set for the pelvis and feet points model. The same applies for the other two key points model. Notice that the losses on the training set are similar for all three models. For the validation set, the losses are oscillating more and the objectness loss tends to rise towards the end of the training.

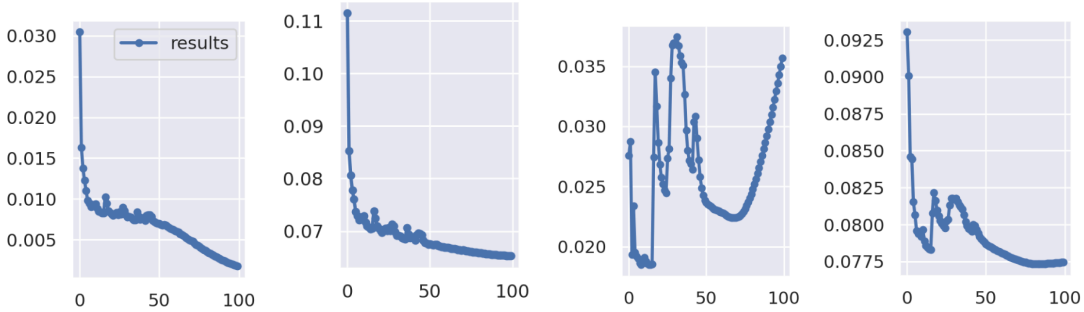
4.1.2 Bounding box model

Figure 4.1.4 illustrates the history of objectness loss and regression loss during training for the bounding box model. Figure 4.1.4a and 4.1.4b are the losses from the training set while Figure 4.1.4c and 4.1.4d are obtained from the validation set. It is worth mentioning that the calculation of each losses were kept the same as in the original YOLOv7 which



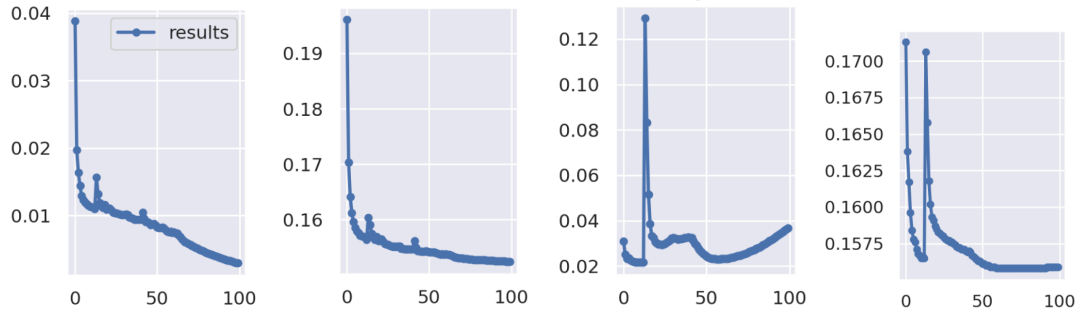
(a) Objectness loss (b) Regression loss (c) Objectness loss (d) Regression loss

Figure 4.1.1: Pelvis and feet points model.



(a) Objectness loss (b) Regression loss (c) Objectness loss (d) Regression loss

Figure 4.1.2: Pelvis and head points model.



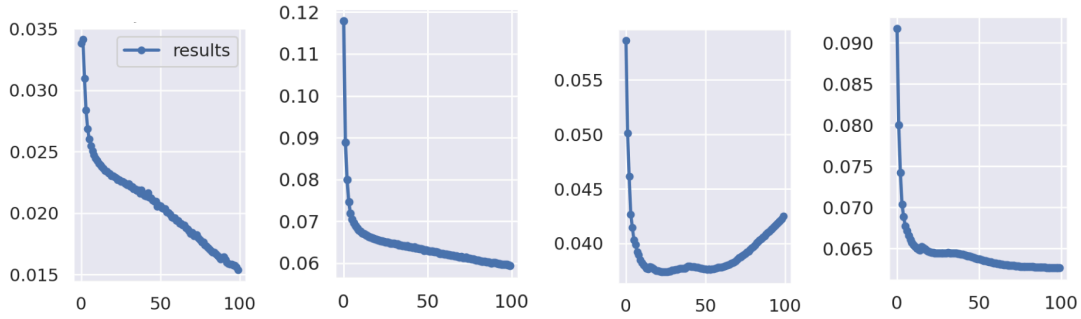
(a) Objectness loss (b) Regression loss (c) Objectness loss (d) Regression loss

Figure 4.1.3: Head and feet points model.

utilizes IoU for objectness.

4.2 Test results on SoccerScene

The four models were tested on a dataset which was created with 10% of the SoccerScene data. This testing dataset has a total of 8000 images and 131231 target labels.

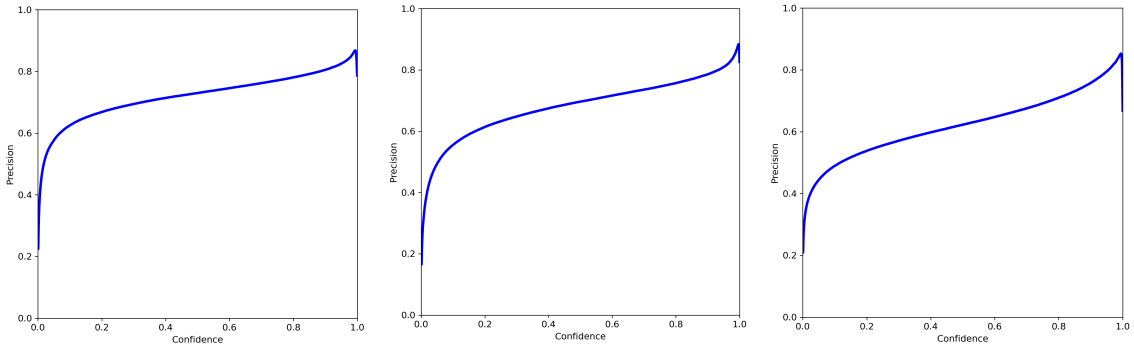


(a) Objectness loss (b) Regression loss (c) Objectness loss (d) Regression loss

Figure 4.1.4: Loss components during training.

4.2.1 Non-Maximum Suppression in image space

The precision, recall and F1 score curves of the key points models are shown in Figures 4.2.1, 4.2.2 and 4.2.3, respectively. A summary of the model performance is given in Table 4.2.1. In the table, the mAP@.1:1.0 denotes the mAP over a range of 0.1 to 1 as the threshold for defining a positive detection using the normalized key points distances. Note that for the bounding box model, the middle point of each box and the middle point of the bottom edge of each box were taken from the model predictions to relate to the best performing key points model. By doing this, the NMS could be done in the same way as the key points models for comparison. Figure 4.2.4 shows the detection performance for the bounding box model with the post-processing part done as mentioned above.

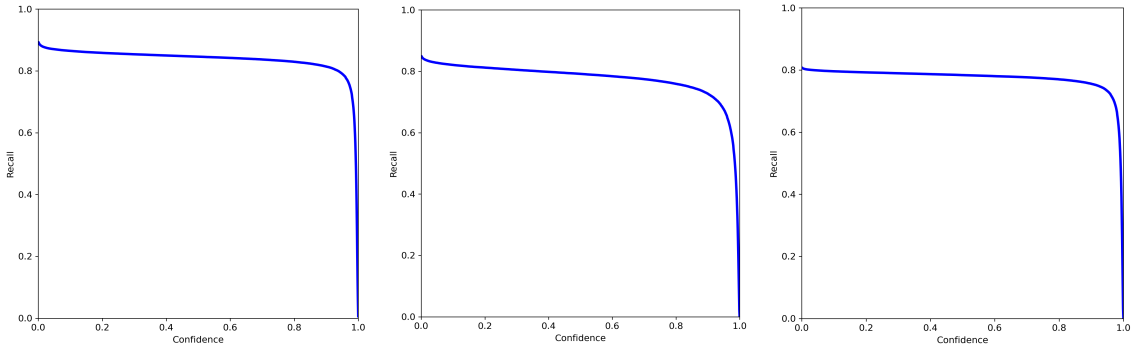


(a) Pelvis feet model (b) Pelvis head model (c) Head feet model

Figure 4.2.1: Precision curves.

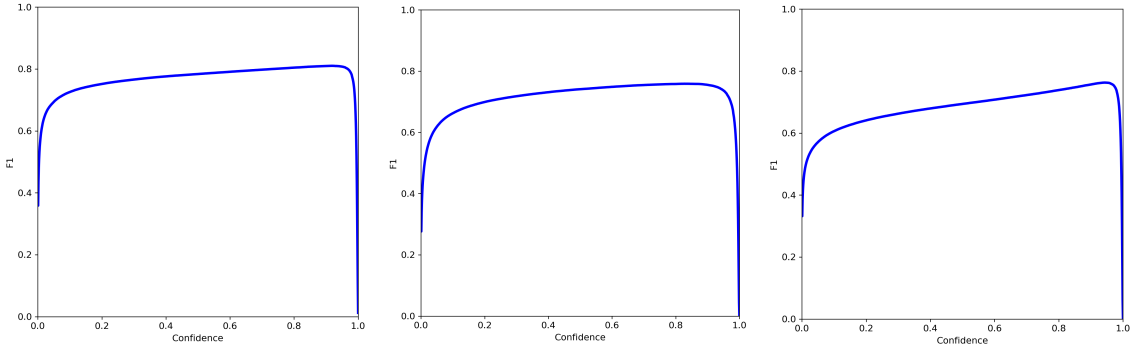
4.2.2 Non-Maximum Suppression in world space

Another way of comparing the models is by performing NMS in the world space. The best performing key points model, which is the pelvis and feet points model was compared with the bounding box model with NMS using the distances between the projected feet



(a) Pelvis feet model (b) Pelvis head model (c) Head feet model

Figure 4.2.2: Recall curves.

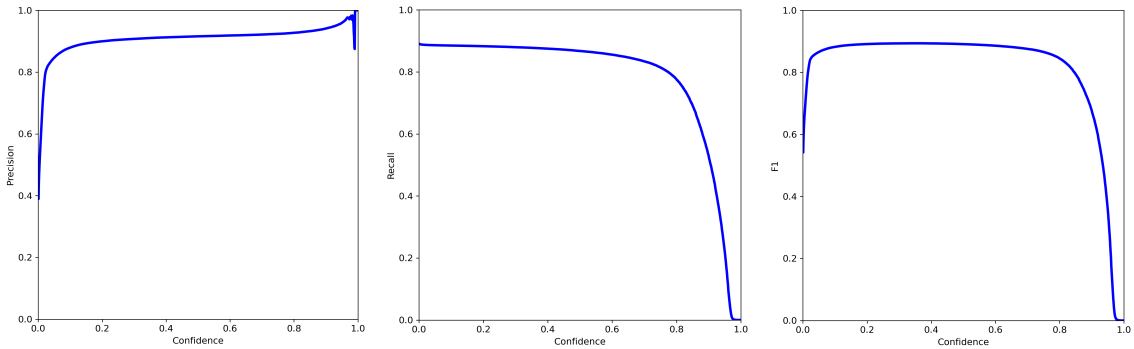


(a) Pelvis feet model (b) Pelvis head model (c) Head feet model

Figure 4.2.3: F1 score.

Table 4.2.1: Summary of model performance with NMS in image space.

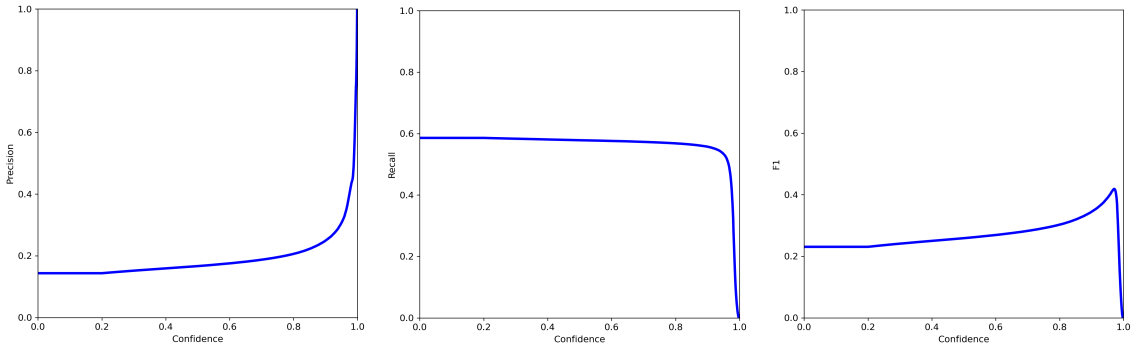
Model	**F1 score**	**mAP@.1:1.0**
Pelvis feet	0.81 @0.918	0.686
Pelvis head	0.76 @0.828	0.625
Head feet	0.76 @0.944	0.598
Bounding box	0.89 @0.341	0.827



(a) Precision curve (b) Recall curve (c) F1 score

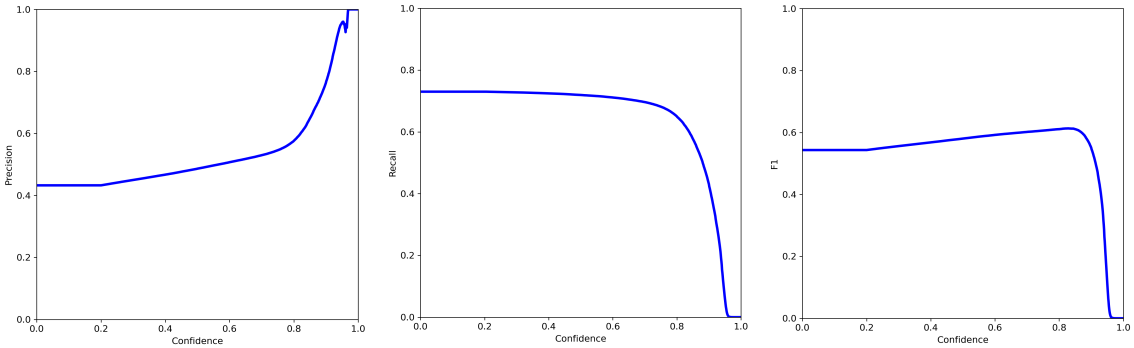
Figure 4.2.4: Detection performance for the bounding box model.

points in world space. Similar to the previous section, the middle point of each box and the middle point of the bottom edge of each box were taken from the bounding box model predictions. From Figures 4.2.5 and 4.2.6, the bounding box model performed better than the pelvis and feet points model. Table 4.2.2 lists the model performance with NMS in world space.



(a) Precision curve (b) Recall curve (c) F1 score

Figure 4.2.5: Detection performance for the pelvis feet model.



(a) Precision curve (b) Recall curve (c) F1 score

Figure 4.2.6: Detection performance for the bounding box model.

Table 4.2.2: Summary of model performance with NMS in world space.

Model	**F1 score**	**mAP@.1:1.0**
Pelvis feet	0.42 @0.971	0.252
Bounding box	0.61 @0.828	0.549

4.3 Test results on SoccerCrowd

The other test case was done on the whole SoccerCrowd dataset. The dataset consists of 4280 images and 292446 target labels. In general, the scenes are more crowded in this dataset as more objects are located in each image.

4.3.1 Non-Maximum Suppression in image space

Table 4.3.1 summarizes the performance of the three key points models and the bounding box model tested on the SoccerCrowd dataset. Similar to the results shown in Section 4.2.1, the pelvis and feet points model performed the best out of the three key points models. The bounding box model is then compared with it using the same way as described in Section 4.2.1. By comparing Figure 4.3.1 and Figure 4.3.2, it can be observed that there are more excessive predictions in the pelvis and feet points model.

Figure 4.3.1: Prediction results from the bounding box model.

Figure 4.3.2: Prediction results from the pelvis and feet point model.

Table 4.3.1: Summary of model performance with NMS in image space.

Model	**F1 score**	**mAP@.1:1.0**
Pelvis feet	0.74 @0.924	0.587
Pelvis head	0.69 @0.848	0.54
Head feet	0.70 @0.949	0.52
Bounding box	0.86 @0.406	0.782

4.3.2 Non-Maximum Suppression in world space

Table 4.3.2 provides the performance comparison with NMS done in world space. The results between the two datasets when doing NMS in world space is not as different as the results obtained from performing NMS in image space. More detailed comparison will be provided in Section 5.2.

Table 4.3.2: Summary of model performance with NMS in world space.

Model	**F1 score**	**mAP@.1:1.0**
Pelvis feet	0.41 @0.971	0.245
Bounding box	0.63 @0.788	0.552

Chapter 5

Discussion

5.1 Evaluation of results

From the results presented in Chapter 4, it can be concluded that the pelvis and feet points model performs the best out of the three key points models. The pelvis and feet points model outperformed the other two key points models in all metrics including precision, recall, F1 score and AP. For the tests done on the SoccerScene dataset with NMS in image space, there are a couple of interesting trends observed from the curves. First in the precision curves, the value of precision for the key points models does not rise as fast as the bounding box model as the confidence threshold increases. This implies that if a low confidence threshold is applied during post-processing, there will be more false positive detections in the key points models than the bounding box model. Looking at the recall curves compared to the bounding box model, the key points models tend to still have a relatively high recall value when the confidence threshold is in the range between 0.8 to 0.95. In other words, the key points models could correctly identify true positives when the confidence threshold is high. Lastly, the F1 curves for the key points models have the highest F1 values at confidence threshold around 0.9 while the bounding box model peaks at a lower confidence threshold around 0.3 to 0.4. It could be interpreted as the key points models have a tendency of producing more high confidence predictions which are not being removed during NMS.

For the tests performed with NMS in world space, all performance metrics had a lower value for both the key points (pelvis and feet) model and the bounding box model. The precision was significantly affected with the values rising much later compared to NMS in

image space. The shape of the recall curves were kept approximately the same but with lower recall values. Discussions about the post-processing parts will be given in Section 5.2.

The SoccerCrowd dataset was created to simulate scenarios of occlusions or when objects are close to each other. It is considered to be the harder dataset for the object detection models. The results reflected this by looking at the evaluation metrics. When comparing the results between the bounding box model and the best performing key points model which is the pelvis and feet points model, the bounding box model has a slight advantage in terms of detection performance.

Even though the bounding box model performs better in terms of detection metrics, the key points models still provide potential benefits for the steps following detection for example tracking. During tracking when two objects are moving towards each other in the image space, the two bounding boxes would often shrink into one when the objects are close to or overlapping each other. The key points representation could provide more information for the tracker to analyze and help identify whether it is the same object in different frames or two different object of the same class.

5.2 Post-processing

During post-processing, the selection of confidence threshold and NMS threshold plays a big role for the final results. In the image space, the proposed method of using normalized key points distances instead of IoU produced better results than using distances in world space. The normalization with object size is key here as the NMS would not be affected by the object size in image space. With the camera angles in the datasets, it can be seen that the objects further away from the cameras are smaller in the images. Since only the feet points are projected into the world space, normalization was not done in the NMS process. Other than normalization, another source of error is projection. As one can imagine when a camera produces a scene from the side view, the projection of objects on the side further away can provide drastically different results depending on the plane of projection. In this thesis project all the feet points were projected to the ground plane of the pitch, which can be not as accurate. This could explain why the detection results were not as good as performing NMS in image space with normalized key points distances.

5.3 Limitations

YOLOv7 was chosen as the model to implement human key points instead of bounding boxes. The working principle behind YOLO to some extent limited the performance of the model. In YOLOv7, the images are divided into a limited amount of grid cells in each detection layer. When the center of objects are located in the same grid cell, which can happen when objects are close to each other, it could be hard for the model to correctly locate these objects when the number of grid cells are low.

When training the key points models, the hyperparameters were kept the same as for training the bounding box model rather than optimized, which can have a negative impact on the model performance. Weights for the loss components for example could affect the final loss values during training hence affecting the weights for the final model. The network architecture in the original YOLOv7 was implemented to optimize the performance of bounding box models, it is unknown that if the structure could produce ideal results when it is performing regression directly only on the location of key points.

Unlike the case of bounding boxes where the IoU threshold in NMS has a general range that gives better results on mAP or F1 score based on previous studies, the key points distance threshold was decided on the performance of the pelvis and feet points model tested on the SoccerScene dataset. Since the threshold is a key factor in the model's final predictions, other thresholds may produce better results when the datasets are different.

For any machine learning model, the training dataset is a crucial factor that affects the performance of the model. In the synthetic datasets, the human objects were rendered with a variety of jersey colors, body size, etc. Notice that in most images, not all the human objects are in poses that are natural or common in football matches such as running. In real life scenarios, when the objects are moving at a high speed, motion blur could negatively impact the model performance. Furthermore, the background of the images were collected from Allsvenskan stadiums with different light conditions. Even though the datasets cover diverse situations, it is worth keeping in mind that factors including lights, weather conditions and camera angles could all make a difference when generalizing the results.

5.4 Future work

The key points models in the thesis project were trained on combinations that consisted of two key points and only the head, pelvis and feet points were considered. Key points combinations with more points from the body could be explored. It would be beneficial to identify the most important key points and come up with a metric that could estimate the poses of the human objects.

The ground truth labels of the datasets used in this thesis project are all the same class which is human, making it a single class classification problem. It could be interesting to see how the key points models perform when there are multiple positive classes. For example with the synthetic datasets, the models could be trained to predict players, referees and bystanders. This could further enable team recognition which not only contributes to the auto-follow process of the cameras but also the collection of more advanced data for analytics.

Chapter 6

Conclusion

In this thesis project, key points models were implemented and trained based on YOLOv7. First, anchor key points were implemented by take corresponding points from the anchor boxes. This changed the problem from regressing the center location, width and height of the box into regressing two key points locations. In the loss function, key points distances were used for objectness and regression loss calculation. Post-processing technique was experimented with NMS. In the image space, key points distances normalized by object size was used for NMS. The feet points were projected onto the ground plane in the world space to perform NMS with distances between the projected points.

The models were tested on two different datasets, with one simulating football match scenarios and the other consisting of hard situations where there are more human objects and more occlusions. The results showed that the model predicting the pelvis and feet points performed the best out of the three key points models while still behind the bounding box model. In both test cases, the NMS in image space produced better results than performing NMS in image space with only one point from each object projected. Although the key points models did not achieve better detection performance than the bounding box model, detection of human key points especially the feet proved to be more valuable than the pelvis and head points.

References

[1] Frevel, N., Beiderbeck, D., and Schmidt, S. L. "The impact of technology on sports–A prospective study". In: *Technological Forecasting and Social Change* 182 (2022), p. 121838.

[2] Randers, M. B., Mujika, I., Hewitt, A., Santisteban, J., Bischoff, R., Solano, R., Zubillaga, A., Peltola, E., Krustrup, P., and Mohr, M. "Application of four different football match analysis systems: A comparative study". In: *Journal of sports sciences* 28.2 (2010), pp. 171–182.

[3] De Silva, V., Caine, M., Skinner, J., Dogan, S., Kondoz, A., Peter, T., Axtell, E., Birnie, M., and Smith, B. "Player tracking data analytics as a tool for physical performance management in football: A case study from Chelsea Football Club Academy". In: *Sports* 6.4 (2018), p. 130.

[4] Mavrogiannis, P. and Maglogiannis, I. "Amateur football analytics using computer vision". In: *Neural Computing and Applications* 34.22 (2022), pp. 19639–19654.

[5] Burić, M., Pobar, M., and Ivašić-Kos, M. "Object detection in sports videos". In: *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. IEEE. 2018, pp. 1034–1039.

[6] Balaji, S. R., Karthikeyan, S., and Manikandan, R. "Object detection using Metaheuristic algorithm for volley ball sports application". In: *Journal of Ambient Intelligence and Humanized Computing* 12 (2021), pp. 375–385.

[7] Hiemann, A., Kautz, T., Zottmann, T., and Hlawitschka, M. "Enhancement of speed and accuracy trade-off for sports ball detection in videos—finding fast moving, small objects in real time". In: *Sensors* 21.9 (2021), p. 3214.

[8] Lu, K., Chen, J., Little, J. J., and He, H. "Lightweight convolutional neural networks for player detection and classification". In: *Computer Vision and Image Understanding* 172 (2018), pp. 77–87.

[9] Zou, Z., Chen, K., Shi, Z., Guo, Y., and Ye, J. "Object Detection in 20 Years: A Survey". In: *Proceedings of the IEEE* 111.3 (2023), pp. 257–276. DOI: 10.1109/JPROC.2023.3238524.

[10] Viola, P. and Jones, M. "Robust Real-Time Object Detection". In: *Proceedings of Second International Workshop on Statistical and Computational Theories of Vision*. Vol. 57. Jan. 2001.

[11] Dalal, N. and Triggs, B. "Histograms of oriented gradients for human detection". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 1. 2005, 886–893 vol. 1. DOI: 10.1109/CVPR.2005.177.

[12] Ren, S., He, K., Girshick, R., and Sun, J. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.6 (2017), pp. 1137–1149. DOI: 10.1109/TPAMI.2016.2577031.

[13] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C. *SSD: Single Shot MultiBox Detector*. 2015. DOI: 10.1007/978-3-319-46448-0_2.

[14] Zhou, X., Wang, D., and Krähenbühl, P. *Objects as Points*. 2019. URL: https://api.semanticscholar.org/CorpusID:118714035.

[15] Law, H. and Deng, J. *CornerNet: Detecting Objects as Paired Keypoints*. 2018. DOI: 10.48550/arXiv.1808.01244.

[16] Zhou, X., Zhuo, J., and Krahenbuhl, P. "Bottom-Up Object Detection by Grouping Extreme and Center Points". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2019, pp. 850–859. DOI: 10.1109/CVPR.2019.00094.

[17] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. "You Only Look Once: Unified, Real-Time Object Detection". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[18] Redmon, J. and Farhadi, A. *YOLOv3: An incremental improvement*. 2018. DOI: 10.48550/arXiv.1804.02767.

[19] Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. "Feature Pyramid Networks for Object Detection". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 936–944. DOI: 10.1109/CVPR.2017.106.

[20] Ultralytics. *YOLOv5: A state-of-the-art real-time object detection system*. 2021. URL: https://docs.ultralytics.com.

[21] Wang, C.-Y., Bochkovskiy, A., and Liao, H.-Y. M. "YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors". In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023, pp. 7464–7475. DOI: 10.1109/CVPR52729.2023.00721.

[22] Wang, C.-Y., Liao, H.-Y. M., and Yeh, I-H. "Designing Network Design Strategies Through Gradient Path Analysis". In: *Journal of Information Science and Engineering* (2023).

[23] Li, Y., Han, Q., Yu, M., Jiang, Y., Yeo, C. K., Li, Y., Huang, Z., Liu, N., Chen, H., and Wu, X. "Towards Efficient 3D Object Detection in Bird's-Eye-Space for Autonomous Driving: A Convolutional-Only Approach". In: *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*. 2023, pp. 2170–2177. DOI: 10.1109/ITSC57777.2023.10422223.

[24] Li, Z., Lan, S., Alvarez, J. M., and Wu, Z. "BEVNeXt: Reviving Dense BEV Frameworks for 3D Object Detection". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2024, pp. 20113–20123.

[25] Yang, L., Yu, K., Tang, T., Li, J., Yuan, K., Wang, L., Zhang, X., and Chen, P. "BEVHeight: A Robust Framework for Vision-Based Roadside 3D Object Detection". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2023, pp. 21611–21620.

[26] Teepe, T., Wolters, P., Gilg, J., Herzog, F., and Rigoll, G. "EarlyBird: Early-Fusion for Multi-View Tracking in the Bird's Eye View". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops*. Jan. 2024, pp. 102–111.

[27] Zand, M., Etemad, A., and Greenspan, M. "Oriented bounding boxes for small and freely rotated objects". In: *IEEE Transactions on Geoscience and Remote Sensing* 60 (2021), pp. 1–15.

[28] Yang, Z., Liu, S., Hu, H., Wang, L., and Lin, S. “RepPoints: Point Set Representation for Object Detection”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019.

[29] Xu, H.-H., Wang, X.-Q., Wang, D., Duan, B.-G., and Rui, T. “Object detection in crowded scenes via joint prediction”. In: *Defence Technology* 21 (2023), pp. 103–115.

[30] Krizhevsky, A., Sutskever, I., and Hinton, G. E. “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems* 25 (2012).

[31] Ultralytics. *YOLOv5 (6.0/6.1) brief summary*. URL: `https://github.com/ultralytics/yolov5/issues/6998`.

[32] Zheng, Z., Wang, P., Liu, W., Li, J., Ye, R., and Ren, D. “Distance-IoU loss: Faster and better learning for bounding box regression”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 34. 07. 2020, pp. 12993–13000.

[33] PyTorch. *BCEWithLogitsLoss*. URL: `https://pytorch.org/docs/stable/generated/torch.nn.BCEWithLogitsLoss.html`.

[34] Ge, Z., Liu, S., Li, Z., Yoshie, O., and Sun, J. “Ota: Optimal transport assignment for object detection”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 303–312.

TRITA – CBH-GRU-2024:335

www.kth.se