REPUBLIC OF TURKEY
OSTİM TECHNICAL UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES
DEPARTMENT OF COMPUTER ENGINEERING
COMPUTER ENGINEERING
MASTER'S PROGRAM

DARTEASY AUTOMATED DART SCORING SYSTEM USING DEEP LEARNING

MASTER'S THESIS

PREPARED BY
ÖMER ALPEREN KOZA

THESIS SUPERVISOR
DR. OSMAN AKIN

ANKARA-2024

THESIS ACCEPTANCE AND APPROVAL

This study, titled “DartEasy Automated Dart Scoring System” and submitted by Ömer Alperen Koza on 16/01/2024, was found successful as a result of the thesis defense on 26/01/2024 and accepted as a Master's Thesis by our jury.

Date of Approval: 26/01/2024

Jury Member:	**Dr. Osman AKIN** Ostim Technical University	______________
Jury Member:	**Dr. Kıvanç DİNÇER** Ostim Technical University	_________________
Jury Member:	**Dr. Hüseyin POLAT** Gazi University	_________________
Jury Member:	**Title Name SURNAME** … University	_________________
Thesis Supervisor:	**Dr. Osman AKIN** Ostim Technical University	_________________
Co-Supervisor:	**Title Name SURNAME** … University	_________________

APPROVAL

I hereby approve that this study, accepted by the jury, fulfills the requirements for being a Master's/PhD Thesis.

…../…../20….

____

Title Name SURNAME
Institute Director

DECLERATION

I hereby declare that I have given Ostim Technical University the permission to archive all or any part of my Master's/PhD thesis approved by the Institute in printed or digital format and make it accessible under the conditions specified below. With this permission, all intellectual property rights other than the usage rights granted to the University will remain with me and the usage rights of all or a part of my thesis for future studies (article, book, license, patent, etc.) will belong to me alone.

I declare and undertake that my thesis is entirely my own work, that I do not violate the rights of others and that I am the sole authorized owner of my thesis. I undertake that I use the copyrighted resources with written permission, which must be used with written permission from their owners, and to submit copies of the permissions to the University upon request.

Within the scope of the "Directive on Collecting, Organizing and Opening to Access The Theses and Dissertations in Electronic Environment"" published by the Council of Higher Education, my thesis is made available on the YÖK National Thesis Center and Ostim Technical University Open Access System, except for the following conditions.

☒ With the decision of the Institute / Faculty Administrative Board, the open access of my thesis has been postponed for 2 years from my graduation date.[1]

☒ With the reasoned decision of the Graduate School / Faculty Administrative Board, the opening of my thesis has been postponed for 6 months from the date of my graduation.[2]

☒ A confidentiality decision has been made regarding my thesis.[3][4]

Date:

Signature:

---

[1] ARTICLE 6(1) In the event that a patent application is made for a graduate thesis or the patenting process is ongoing, upon the recommendation of the thesis advisor and the approval of the graduate school department, the graduate school or faculty executive board may decide to postpone the access to the thesis for two years.

[2] AR 6(2) For theses that use new techniques, materials and methods, that have not yet turned into articles or are not protected by methods such as patents, and that contain information and findings that may create unfair gain for third parties or institutions if shared on the internet, the thesis may be prevented from being accessed for a period not exceeding six months upon the recommendation of the thesis advisor and the approval of the institute department, with the reasoned decision of the institute or faculty board of directors.

[3] ARTICLE 7(1) The decision of confidentiality regarding graduate theses related to national interests or security, security, intelligence, defense and security, health, etc. is made by the institution where the thesis is conducted. The confidentiality decision regarding graduate theses prepared within the framework of a cooperation protocol with institutions and organizations is made by the university board of directors upon the recommendation of the relevant institution and organization and the approval of the institute or faculty. Theses for which a confidentiality decision is made are notified to the Council of Higher Education.

[4] ARTICLE 7(2) Theses for which a confidentiality decision has been made are kept by the institute or faculty within the framework of confidentiality rules during the confidentiality period, and are uploaded to the Thesis Automation System if the confidentiality decision is lifted.

OSTİM TECHNICAL UNIVERSITY

INSTITUTE OF NATURAL AND APPLIED SCIENCES MASTER'S / DOCTORAL THESIS ORIGINALITY REPORT

Thesis Title: DartEasy Automated Dart Scoring System

Student Name Surname: Ömer Alperen Koza

Thesis Supervisor Title Name Surname: Dr. Osman Akın

Department: Computer Engineering

Program: Computer Engineering

Date: 23 / 02 / 2024

The part of my Master's thesis, which consists of Introduction, Main Chapters and Conclusion and consists of 40 pages in total, was examined by my thesis advisor and myself on 23/02/2024 by the Turnitin plagiarism detection program.

According to the originality report, the similarity score of my thesis is 4%.

Applied filters:

1. Bibliography (excluding)

2. Citations (excluding)

3. Parts of text with less than five (5) words overlapping (excluding)

I declare that my thesis does not contain any plagiarism; that I accept all kinds of legal responsibility that may arise in the event that the contrary is detected and that the information I have given above is correct.

Student Signature:………………….

APPROVAL

Date: 23 / 02 /2024

Dr. Osman AKIN

ETHICAL DECLERATION

I declare that this study is an original study, that I act in accordance with scientific ethics and rules in the preparation, data collection, analysis, presentation of information and all other stages of the study, that I have obtained all the document information in the study within the framework of academic ethics and rules, that I have presented all visual, audio, and written information and results in accordance with the scientific rules of ethics, that I have not made any falsifications in the data I have used, that I have referred to the sources I have used in accordance with scientific norms, that my thesis has been written by me and is original, except for the cases where I cited sources, that it has been produced by me under the supervision of my thesis advisor Dr. Osman AKIN and written in accordance with Ostim Technical University thesis writing guide.

Student Signature

Ömer Alperen Koza

23 / 02 / 2024

ACKNOWLEDGEMENTS

I want to express my gratitude to my advisor Dr. Osman AKIN for his invaluable patience and feedback. This work cannot be done without his motivation and help. I would also like to thank to my professors in the defense committee, Dr. Kivanc DINCER and Dr. Huseyin POLAT who generously provided their knowledge and expertise. Additionally, I would like to mention my appreciation to my Faculty members for their support during my journey.

23 / 02 / 2024

Ömer Alperen KOZA

ÖZ

Yazar Adı ve Soyadı : Ömer Alperen Koza

Üniversite : OSTİM Teknik Üniversitesi

Enstitü : Fen Bilimleri Enstitüsü

Program Adı : Bilgisayar Mühendisliği

Tezin Türü : Yüksek Lisans Tezi

Sayfa Sayısı : 40

Tarihi : 2024

DARTEASY DERİN ÖĞRENME İLE OTOMATİK DART SKORLAMA SİSTEMİ

Dart oyununda skor tutma işlemi geleneksel yöntemlerde insan hesaplamalarına dayanır ve dolayısıyla hatalara ve tutarsızlıklara açıktır. Bunun yanında, oyunculara gereksiz ve stresli bir iş yükü yüklemekte ve bir kişiyi skor tutma işiyle meşgul etmektedir. Bu problemin çözümü için elektronik dart ya da otomatik okuma sistemleri önerilmiştir. Ancak bu çözümler pahalı, yavaş ya da bazı durumlarda istenilen doğru sonuçları vermemektedirler. Bunun yanında, kalabalık mekanlarda uygulanması zordur. Bu yöntemler çoklu kamera ya da tekli kamera çözümlerini içerir. Her ne kadar halihazırda derin öğrenme tabanlı yöntemler olsa da bu çözümler de başarı olarak istenilen düzeyde değildir. Bu başarısızlık nesne algılama algoritmalarının dart oyununda yeterli seviyede olmamasından ya da bizzat kullanılan yöntemin kendi yetersizliğinden kaynaklanmaktadır. Bu tezde, "referans noktası" dediğimiz bir noktayı referans alarak dartların konumlarını tespit etmek için anahtar nokta tespit teknolojisine dayalı olarak ucu otomatik olarak tespit edecek bir derin öğrenme sistemi ve çeşitli yöntemler önerilmiştir. Başlangıçta, algoritmayı daha hızlı hale getirmek ve daha iyi sonuçlar elde etmeye yardımcı olmak için görüntülerin arka alanını kırpıyoruz ve eğitim için yalnızca değerli parçaları kullanıyoruz. Daha sonra dartı tespit etmek ve ucunu bulmak için üç farklı yöntem öneriyoruz. Ana modelimizde hedefimiz sınır kutusunun içine bir üçgen çizip, üçgenin kenarının konumunu değiştirerek dartın ucunu bulmaktır. Yaklaşımımızı farklı veri kümeleri üzerinde denedik ve iki yöntemin 99% başarı oranına ulaştığını gösterdik

Anahtar Sözcükler: Derin öğrenme, dart oyunu, skorlama, nesne tanıma, segmentasyon

ABSTRACT

Thesis	: Ömer Alperen Koza
University	: OSTİM Technical University
Institute	: Graduate School of Natural and Applied Sciences
Program’s Name	: Computer Engineering
Thesis Type:	: Master
Pages	: 40
Year	: 2024

DARTEASY AUTOMATED DART SCORING SYSTEM USING DEEP LEARNING

Traditional methods of keeping score in dart games rely on human calculations and are therefore prone to errors and inconsistencies. In addition, it puts an unnecessary and stressful workload on players and keeps one person busy with keeping score. Electronic darts or automatic detection systems have been proposed to solve this problem. However, these solutions are expensive, slow, or in some cases do not provide the desired results. Besides, it is difficult to apply in crowded places. These methods include multi-camera or single-camera solutions. Although there are already deep learning-based methods, these solutions are not at the desired level of success. This failure is due to the object detection algorithms not being at a sufficient level in the dart game or the inadequacy of the method used itself. In this thesis, we proposed a deep learning system and several methods to detect the tip automatically based on keypoint detection technology to detect the darts' positions by referencing a point which we call the “reference point”. Initially, we crop the images and use only the valuable parts for training to make the algorithm faster and help to achieve better results. After that, we propose three different methods for detecting a dart and finding its tip. Our main model is to draw a triangle inside the boundary box and find the tip of the dart by changing the position of the triangle’s edge. We experimented our approach on different datasets and showed that two methods have reached the 99% success rate

Keywords: Deep learning, dart game, scoring, object detection, segmentation

İÇİNDEKİLER

LIST OF TABLES

LIST OF FIGURES

SYMBOLS AND ABBREVIATIONS

A.I.	Artificial Intelligence
WDO	World Darts Federation
YOLO	You Only Look Once
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
CNN	Convolutional Neural Network
R-CNN	Region-based Convolutional Neural Network
OBB	Oriented Bounding Box

1. INTRODUCTION

Dart is known as a "bar game" for most people and is popular mostly in bars and pubs [1]. However, it is now a globally known sport with national and international tournaments organized by "World Darts Federation" [2]. The World Darts Federation has 70 members from all over the world [2].

With the rise of machine learning, image processing has become a main topic. Today, machine learning and image processing are used in a very wide area of applications. They are used in many fields like medical, agricultural, and machinery industries as well as in sports. Many sports branches use machine learning with image processing to calculate the scores or player movements to make correct decisions. They are newly started to be used in "darts" too.

Many different darts games are using the same board and same steel-tip darts. These games can be played one-on-one or team vs. team. Every game has a different type of scoring system but all of them are based on the position of the darts. A dart board is divided into 62 different areas and the points of the player are determined which one of them is hit during the game. Some game types rely on the number itself which is hit, and some game types count the value of the number. All scores must be kept by the players (or someone dedicated just to keep scores) during the game (Figure 1). This way of computing is open to errors and lessens the joy of playing. Sometimes it is not just reading the numbers but also doing some calculations. For example, one of the most popular dart game types X01 needs to reduce every scored number from a predetermined number like 301, 501, or 1001. For some non-professional organizations, it can be a problem.

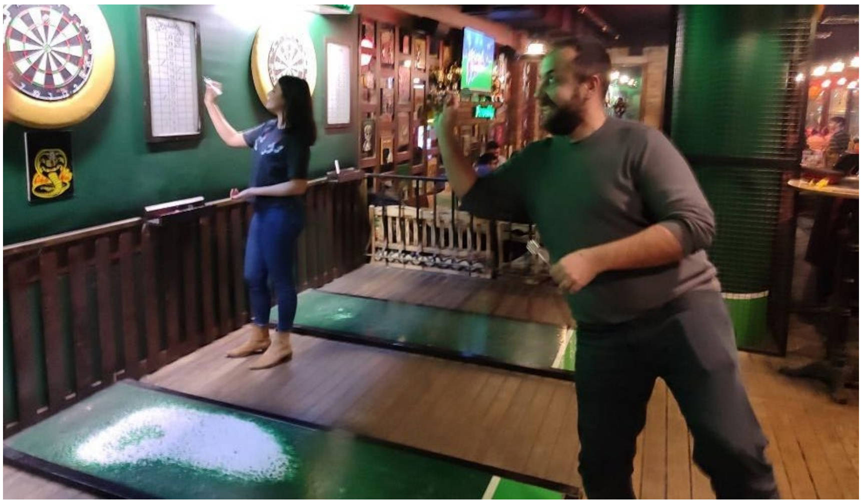

Figure 1. A photo from a public tournament. Scorekeepers can be seen near the board.

To overcome the score computing problem, many different ways have been used. One of them is electronic dart boards with plastic darts (Figure 2). This one is used in homes but it is not preferred in public places and tournaments. For steel-tip darts, some solutions have been found and started to be used as commercial products (Figure 2). The products use object detection for computation. All of them use several cameras around the dart board. They mainly use a light ring around the board and place the cameras (generally 3 cameras) around this light ring. Although their success in correct computations, they have some disadvantages too. The first deficit is the need for calibration. Secondly, they are limited to operating with limited brands of dart boards. The third deficit is the price. During our research, we could not find much academic study about this kind of multi-camera system.

Figure 2. Electronics dart board (left photo) and Scolia automatic dart system (right photo)

As our purpose is to develop a system that is more accessible and can be used in every event, we tried to build a system with a single camera. Since our primary aim is to be able to use the system everywhere like homes, bars, tournaments, etc. we decided to place the camera at a fixed position with a fixed angle. This is necessary because a portable camera is dangerous for itself, the players, and the audience. According to this approach, placing the camera on a light ring like the currently used ones is a good idea. However, this needs to have extra hardware, which is the light ring. Instead of this, we place the camera in a far position from the board, facing towards the board with a small angle. This can make the predictions more accurate and do not affect any movement of the players.

Object detection systems improved exponentially over time. Many algorithms have been developed and used. RNN (Recurrent Neural Network) and one of its branches LSTM (Long Short-Term Memory) are two of them. These algorithms are good at sequential data where you might predict the sequence.

CNN (Convolutional Neural Network) is another algorithm which is widely used in deep learning. R-CNN (Regional Convolutional Neural Network) is a version of this algorithm especially used in object detection. To make this algorithm faster, it is developed and two versions have appeared, Fast R-CNN and Faster R-CNN.

In this paper, we used a deep learning system using keypoint detection technology and anchor-free YOLOv8 algorithm to detect the darts' positions by referencing a point which we call the “reference point”. YOLOv8 claims that it is the fastest algorithm for object detection and tracking.

We first crop the images and use only the valuable parts. This makes the algorithm work faster and helps to achieve better results. After that, we propose three different methods for detecting a dart and finding its tip. Two of them are experimented and the success rates are above 99%. We used different datasets and combined them for the purpose.

Our contributions are: We made two main contributions to automatic dart scoring systems. One of them is about calculation speed. We applied an automatic cropping to remove the area which does not have meaning to us. This removed area is the area other than the dartboard. By extracting the background from the images on the circle boundaries, we decreased the training and detecting computation time. Another contribution we made is to detect the dart as a whole. By this approach, we increased the accuracy. Furthermore, we proposed three new methods in this paper to find the dart tip. It is essential to find the score. All of the methods are related to boundary boxes which surround a dart. First approach is to find the dart tip according to the area of the bounding box. The tip is either on one of the edges or on the middle point. Another method is to use an oriented bounding box which is drawn with an angle according to the angle of the object. This time the tip is always in the middle. Our main model is to draw a triangle – which is a very close shape to a dart- inside the boundary box and find the tip of the dart by changing the position of the triangle’s edge. Dart’s tip will be on one of the edges of the triangle.

The paper is organized as follows: In section 2, recent studies and implemented works are reviewed. In section 3, we propose three different methods for dart detection and finding its tip to have the correct score. Section 4 is the experiments section where we explain the test results and comparisons between previous works and ours. In section 5, we make a summary as a conclusion and discuss new directions for future research.

2. RELATED WORKS

Our aim is to detect darts on the dartboard. We have used an object detection algorithm for this purpose. Object detection is a machine learning technology which can detect certain objects in images or videos. Like all other machine learning technologies, there are many algorithms to train the computer and make it detect the objects. Each algorithm has its own strong and weak points according to the application.

2.1. Main Object Detection Algorithms

Some main algorithms are explained in the following.

2.1.1. Convolutional Neural Network (CNN)

Deep neural networks have been started to use in many fields and its popularity is getting higher. There are several deep neural network architectures and CNN is one of the most popular ones. In CNN, there are matrices and some linear operations occur between them, which are called convolutions (Figure 3). CNN has multiple levels, which are: convolutional layer, non-linearity layer, pooling layer and fully-connected layer. CNN is one of the most successful algorithms for image recognition, voice recognition, image tracking etc. [3].

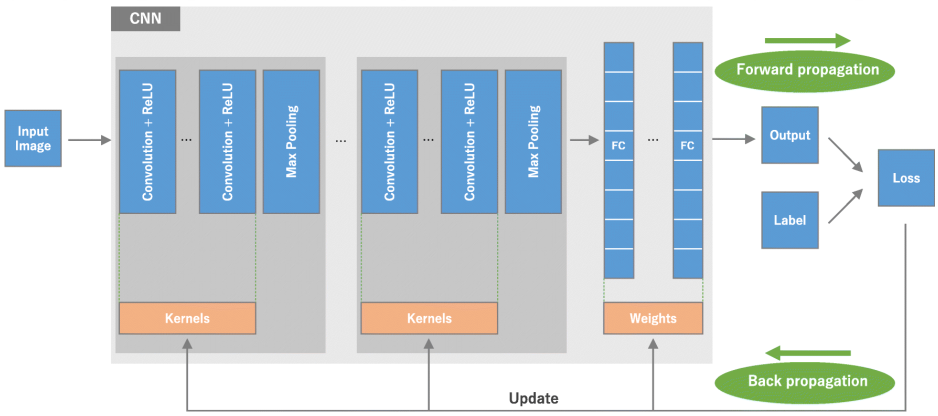

Figure 3. An overview of CNN architecture and the training process [4]

Convolutional Layer: Convolution operations are applied on the input to get some filtered results for future usage. These linear operations are applied to some part of the input and consist of an array of numbers [4]. Input is a matrix, consisting of numbers which are generally color codes. The operational matrix is smaller than the input and is applied part by part. Every operation returns a single value. These single values also create a new matrix. For example, if a 5x5 input (we take the input as 2-dimensional for simplicity) is convolved by 3x3 matrix, the resulting matrix would be a 3x3 matrix because the filtering matrix is applied to a region and swipes 1 column and applied again (Figure 4 and Figure 5).

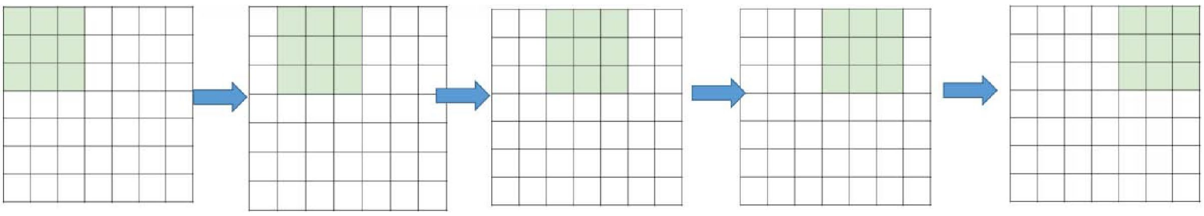

Figure 4. Filtering matrix travels along the input matrix. It first swipes one column and when it reaches the last column, it returns to the first column and swipes 1 row below. This operation continues till all input matrices are covered. Every operation result is a value and these values are stored in another matrix. For example, in this figure, the input is a 7x7 matrix and the filter is a 3x3 matrix. The result is a 5x5 matrix [3].

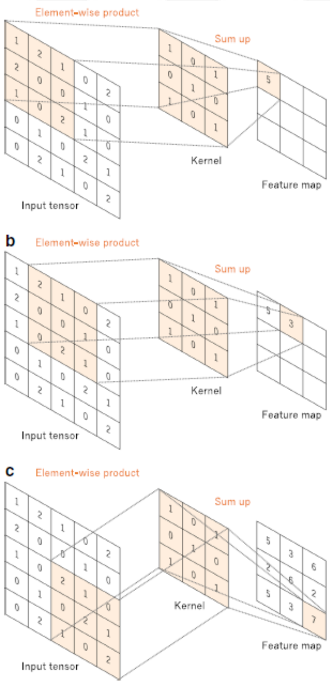

Figure 5. An example of convolution. [4]

There might be more than one filter, this time the convolution number increases. Sometimes "padding" is applied which makes the result matrix's dimension the same as the input. This operation adds enough columns at the beginning and the end as well as enough rows at the top and one bottom full of zeros. Here, enough means the number of rows and columns to match the sizes of the output and the input.

Non-linearity Layer: After the linear operations in the convolution layer, to adjust the previously generated output, a non-linear operation is applied. In many applications, linearity may not be enough to explain the data. Thus, non-linearity is needed. This layer is also called an “Activation Layer” because in this layer, an activation formula is applied to the data. This non-linear activation formula is used to be one of the following formulas: sigmoid and hyperbolic tangent (tanh). The reason behind using these functions is the similarity between these formulas and biological neuron behavior [4]. However, when the neural network gets deeper, the gradient signals of these functions become close to zero. This makes the signal vanish. To overcome this problem, a new function “Rectified Linear Unit (ReLU)” has started to be used recently (Figure 6). ReLU does not vanish and it is much simpler [3].

$$\mathrm{ReLU}(x) = \left\{ \begin{array}{l} x, \text{ if } x >= 0 \\ 0, \text{ if } x < 0 \end{array} \right\} \tag{2.1}$$

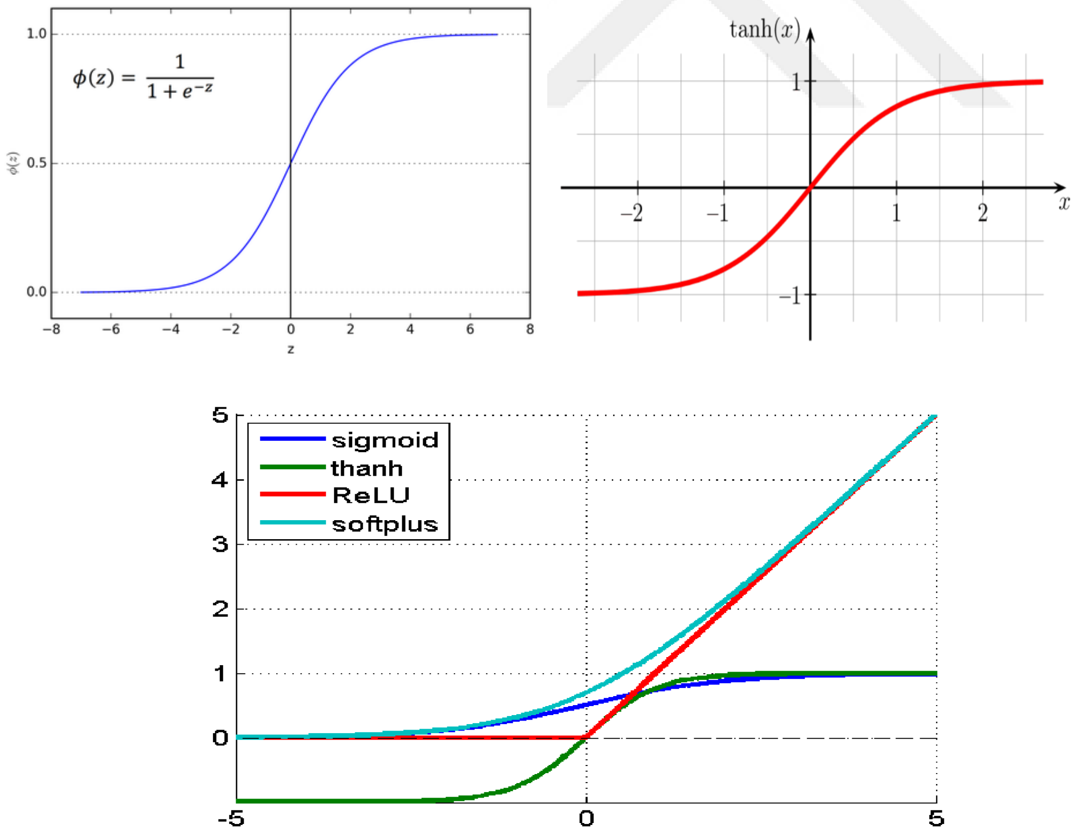

Figure 6. Sigmoid and tanh functions (first and second images). Comparison of activation functions (third image).

Pooling Layer: Pooling is used to reduce the complexity of future operations by down-sampling [3, 4]. A frame with a predetermined size is put onto the layers created previously. After that, generally the maximum value (max pooling) or the average value (global average pooling) of the values inside that frame is calculated and that value will be an entry for the next neuron (Figure 7). Frame travels all values and with the result, a new matrix is created.

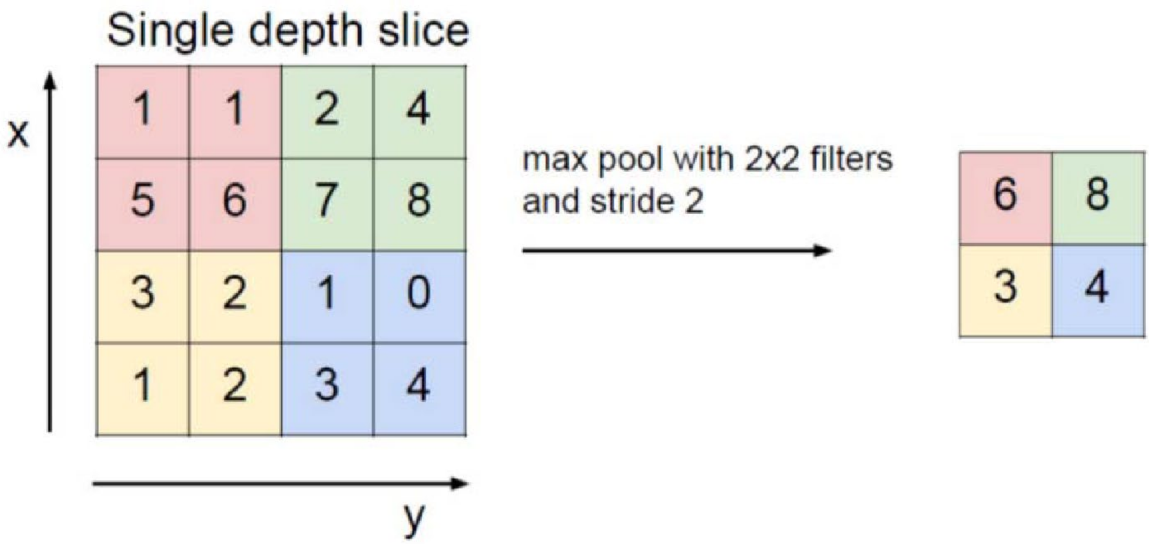

Figure 7. An example of Max Pooling. [3]

Fully Connected Layer: This layer is the last layer and at the end we get the final results (Figure 8). At the beginning, all results in the previous layer are flattened, meaning all values are converted into a one dimensional matrix. The name “fully connected” comes from the connections between neurons. Each neuron applies a linear transformation with a weight matrix. All possible connections are present between the neurons in the fully connected layer.

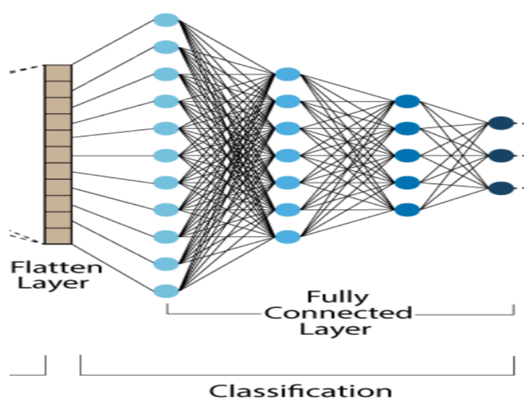

Figure 8. A representation of a fully connected layer. Flatten layer at the entrance is the flattened data matrix.

Images are complex and large inputs with too many parameters. With traditional ways, training would be very hard since every neuron is connected to each other. The ability of sampling and operating with smaller pieces makes CNN the perfect algorithm for image recognition. Other deep learning algorithms such as RNN (Recurrent Neural Network), need sequential data. RNN can give perfect results if the input is a sequence of meaningful data. However, image and video inputs are not sequential. That makes RNN and other similar algorithms behind CNN for image detection and recognition. There are several CNN-based algorithms widely used in image recognition.

2.1.2. Region-Based Convolutional Neural Network (R-CNN)

As its name stands, R-CNN, starts with finding regions in an image. These regions are proposals which may include an object. After that, it applies CNN procedures to these regional proposals. At last, it classifies the objects (Figure 9) [5].

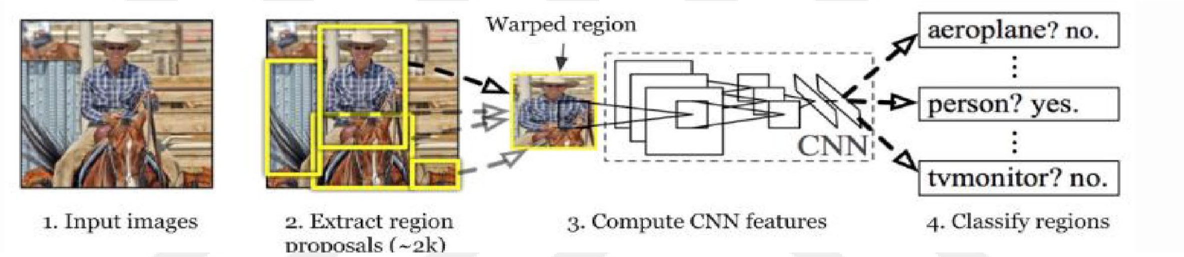

Figure 9. R-CNN architecture [5]

R-CNN, uses a method called "selective search" to determine bounding boxes or region proposals. Selective search tries to combine similar, adjacent points to create regions. The similarity can be color, texture or intensity. After determining the possible regions, R-CNN converts it to a rectangular shape. Next step is to put these regions into the CNN process for image classification. The difference in CNN process is the Support Vector Machine (SVM) which is put to the last layer to detect the object. SVM puts specific linear support vectors. Region proposals are determined by comparing negative and positive values [5, 6].

Although the new, fresh ideas that R-CNN brings to image processing and its better efficiency over traditional methods, it has some weak points. First of all, selective search is a slow process. R-CNN applies selective search to thousands of regions. Besides its slowness, R-CNN occupies too much memory. Because of these drawbacks, newer algorithms have been developed.

2.1.3. Fast R-CNN

The world continues to change and image detection needs increase rapidly. The drawbacks of R-CNN and the increasing need for faster algorithms pushed engineers to develop new algorithms. Fast R-CNN is one of them. Fast R-CNN is faster, more accurate and occupies less memory according to R-CNN. The main difference between Fast R-CNN and R-CNN is the sharing of same computations for overlapping regions. This makes Fast R-CNN more efficient than R-CNN.

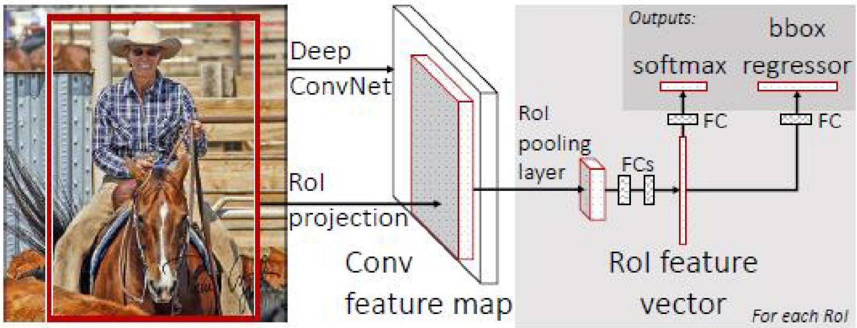

Figure 10. Fast R-CNN architecture [5]

The “region of interest (RoI)” pooling layer in Figure 10, gets the characteristics of the objects and converts them into smaller pieces for the sake of speed. To achieve this, RoI uses max pooling. As, again, can be seen in Figure 10, instead of using different processes for image feature extraction (CNN) and classification (SVM), we now have just one network [6].

2.1.4. Faster R-CNN

Fast R-CNN is much faster than R-CNN, however, it still uses selective search to create the regions and selective search is a slow process. Researchers thought that already a CNN algorithm is used to extract the features of the image. These features are the data to determine the regions. So, for regional proposals, this operation can be used. Two achievements (feature extraction and region proposal) can be done with one process [6]. One CNN training is enough for both of the tasks and this saves nearly all the time spent for region proposals in Fast R-CNN (Figure 11).

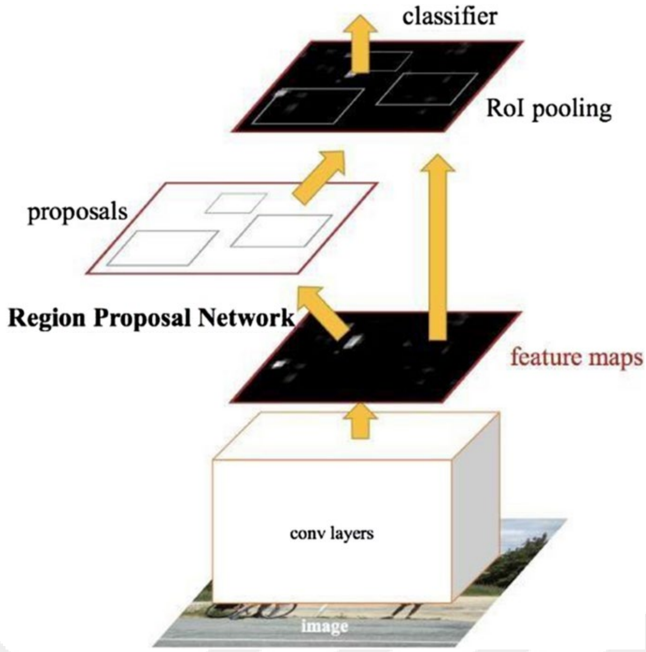

Figure 11. Faster R-CNN architecture.

2.1.5. You Only Look Once (YOLO)

One of the newest and most used image detection and recognition algorithms is YOLO. YOLO has been developed over time and many versions – 1 through 8- have been published.

YOLOv8 is the newest version of the YOLO series at the time of writing this paper. YOLOv8 is developed by Ultralytics [7] and their approach is different from many of the object detection algorithms, even from previous YOLO versions.

YOLO (You Only Look Once), is developed by Joseph Redmon et al. [8], and the main aim of this algorithm is to detect objects very fast with high accuracy. The best part of YOLO is its one-stage object detection model. Many of the other CNN models use double-detection and this makes them slower.

Over time, many updates have been developed to the YOLO algorithm [9] (Figure 12 shows the development of YOLO series). The difference of YOLOv8 is its anchorless architecture. Without having anchor boundaries, an algorithm can detect objects faster and these objects can easily be converted to keypoints [9].

The speed and efficiency difference of YOLO versions can clearly be seen in Figure 17.

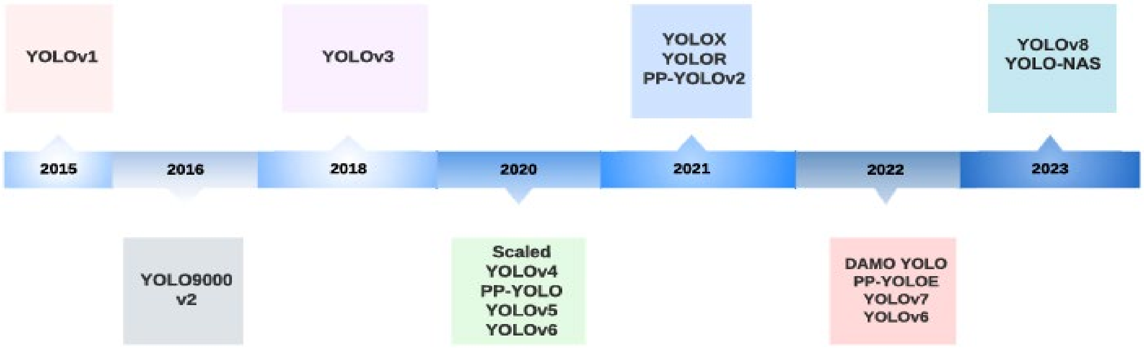

Figure 12. Yolo series chronology.

YOLO's difference is: YOLO is a "single-stage" algorithm while R-CNNs are two-stage. YOLO does not compute the region proposal extraction stage [10]. YOLO basically uses one, relatively simple CNN to predict the bounding boxes. Yolo takes the entire image, predicts bounding boxes and determines the classes with their probabilities (Figure 13) [8, 12].

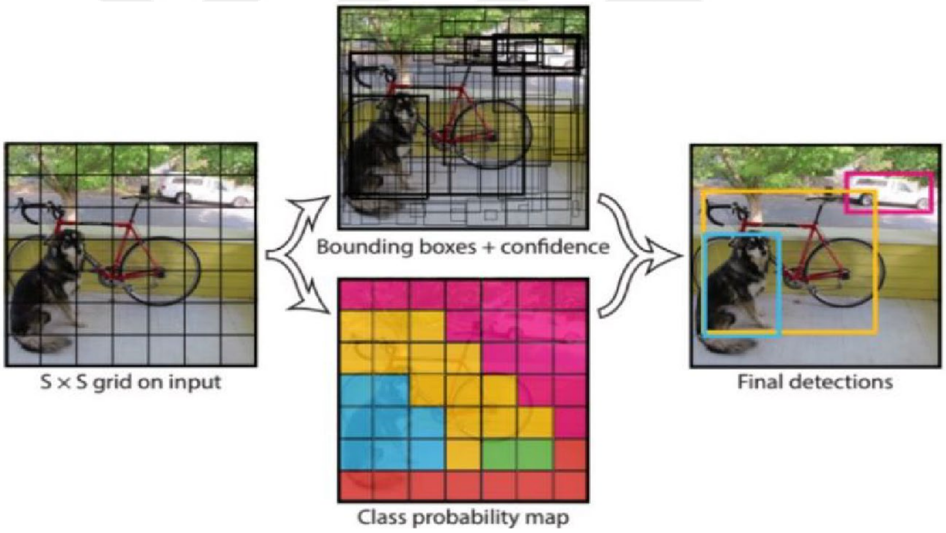

Figure 13. Yolo process.

Basic YOLO working principle consists of the following steps:

- First, YOLO divides the image into grids which consist of equally dimensioned regions.
- These grids try to detect the object individually. The grids make the predictions with a confidence score. The higher the score the more possible the prediction is true. For example, if there is no object detected inside the grid, the confidence score is zero [8].

- After that, all grids predict the bounding box coordinates relative to grid cells.
- After that, a filter (Non-Max suppression) is applied to the boxes. This application filters out overlapped boxes and repeated predictions
- Finally, intersection over union (IoU) is calculated between the predicted box and ground truth box [8, 6, 11, 12].

The difference of YOLOv8: YOLOv8 is the latest and the fastest YOLO algorithm. Although the basics of all YOLO versions are similar, YOLOv8 has a difference which makes it faster. YOLOv8 is an anchor-free algorithm.

Anchor-based method: Most of the CNN algorithms, like R-CNN, Fast and Faster R-CNN, and some YOLO versions, use this method. In this method, there are predefined anchor boxes. These anchor boxes are created like a grid and used as a template for object predictions. Prediction of the object's location and shape is done by referring to the anchor boxes. Anchor-based systems determine the bounding box by using an offset value from an anchor box [13].

Anchor-free method: With anchor-based algorithms, the intersection between the anchor box and ground truth should be calculated during training [14]. However, with the anchor-free method, there is no need for such a process. In the anchor-free method, points of the image are used instead of boxes. The position and the size of an object are calculated according to a given point [13]. Anchor-free model has two branches for calculation: one is to determine if the point belongs to a category and the other is to find the centre point and the offset of the bounding box [14] (Figure 14 and 15).

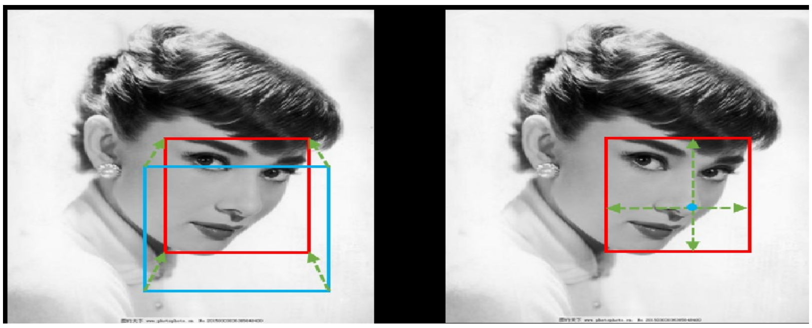

Figure 14. Anchor-based vs. anchor-free. Left is an example of anchor-based and right is anchor-free model. Red boxes are ground truths. Blue box on the left image is the anchor box. Green lines are the offsets. On the left image, the offset is predicted based on the anchor box. On the right image, on the other hand, offset is directly estimated from the taken point [13].

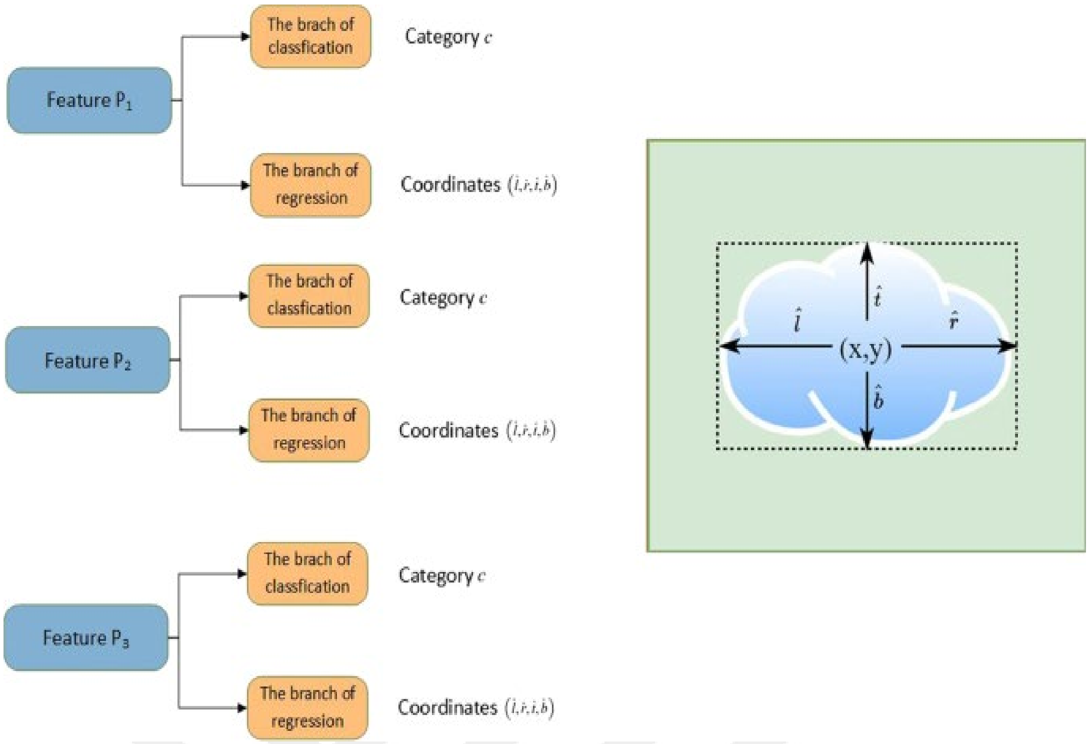

Figure 15. Anchor-free detector's structure [14].

Using an anchor-free method, YOLOv8 (Figure 16) becomes faster than other YOLO versions (Figure 17). Also taking the processes of object prediction, classification and regression independently, allows each branch to handle its own tasks. This is a big factor in increasing the accuracy. To calculate the probability of having an object inside of a bounding box, YOLOv8 uses sigmoid function. For class probabilities, it uses softmax [9].

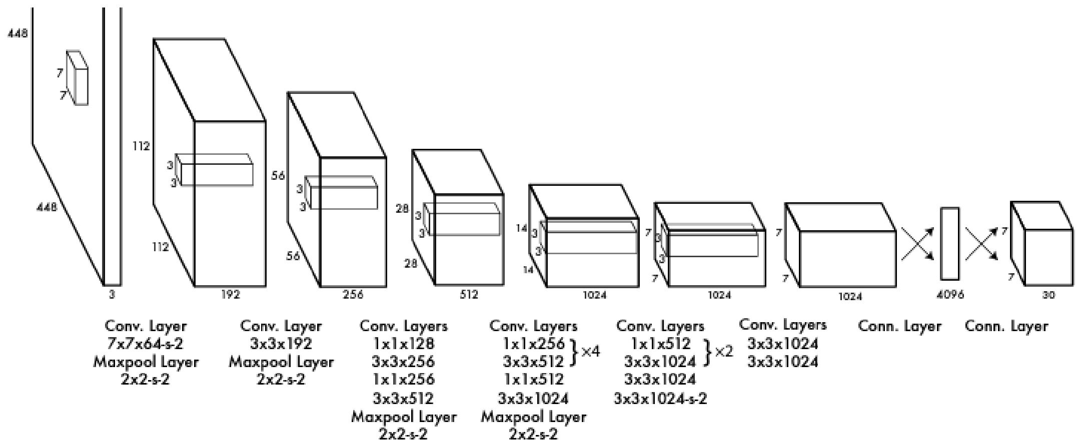

Figure 16. YOLOv8 architecture [8].

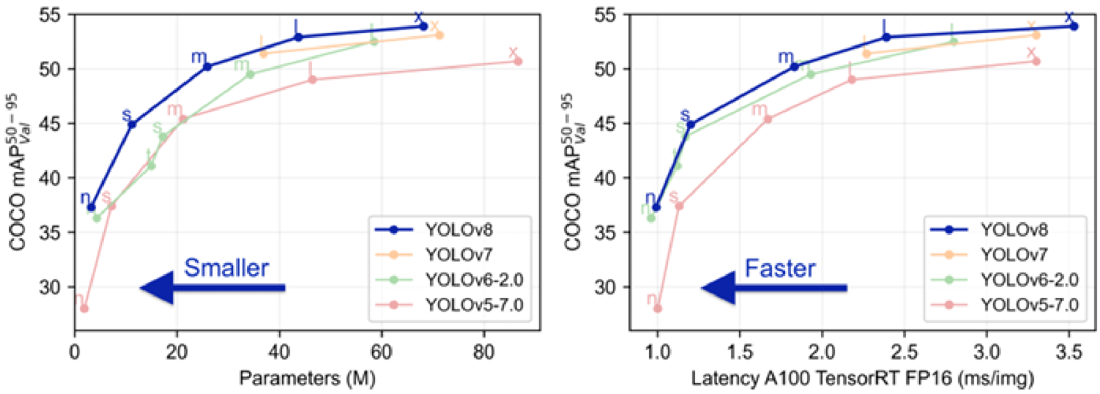

Figure 17. Speed comparison between YOLO versions [15]

2.2. Studies on Automatic Dart Scoring

During our research, we found that the academic studies on this particular topic are far from expected. There is only one published work which is targeting automated dart scoring [4]. William McNally et al. [16] use a single camera with a keypoint system to detect the positions of the darts. There are commercial applications that claim to score automatically but we were unable to find academic works belonging to them. In their "deep dart" system, McNally et al. use a single camera whereas the commercial ones use three cameras. McNally's work explains the system mathematically and comes up with a new idea. They think that 3-camera systems are not easily accessible and having a one-camera system can be used everywhere with low costs. The authors find the tips of the darts which stick into the board as objects and use them like keypoints. They define four calibration points and with the help of these points, they find the coordinates of the dart tip. Using the distance and angles of this point to the intersection of four calibration points, they can calculate the point that the dart hit. A CNN-based algorithm has been used with YOLOv4 for object detection. The model has been trained by a dataset of 16.000 images.

Roman Martsyshyn et al. propose a different approach to dart score reading [17]. They use infrared sensors to sense the dart and its landing position. With this approach, there is no object detection but extra sensors are needed.

Besides the published academic studies, there are some practices published online. One of them is *"opencv-steel-darts"* by Hannes Hoettinger [18]. In this study, the author uses the opencv library of Python to detect the darts. Deep learning is not the core of these studies. The success rates are not very high but not very low either. However, the computation process is very slow.

Another open-source work is *"Darts_Project"* by Lars Gudjons [19]. In this project, Gudjons compares 2 images continuously and takes the difference -which is the newly thrown dart -. After having the difference, they try to get a triangular shape which is close to a dart shape. One of the edges is the pin point which touches the board and by using coordinates to a reference point, they calculate the position of the dart.

In all of the published works mentioned above, there are also some commercialized works. Some companies like Scolia [20], Autodarts [21], Dartsee [22] and Target [23] have built their auto-scoring systems. These systems mainly use 3 cameras to apply object detection.

Besides these works, there are similar studies which are not specifically about dart scoring but deep learning and object detection of similar items. Blowgun game scoring [24] and archery scoring systems [25] are examples of these kinds of works. However, in these works, the authors use edge detection methods to find the intended object. Deep learning, on the other hand, is a newer, more capable and more adaptable system.

Deep learning is an approach of modeling, which uses many layers to understand the data. These models have improved the "state-of-art" in many fields such as object and sound detection and recognition. Misra et al. and LeCun et al. give us information about deep learning [24, 27]. Voice recognition, face detection, object detection and tracking are only some of the numerous applications that use deep learning. We encountered many works on deep learning during our research and some of them are cited in this paper.

There are many algorithms used in deep learning. We will explain them in detail in Section 3. As mentioned in Section 1, RNN and LSTM, which is also an RNN-based algorithm, are not suitable for us because of the lack of a sequence. In a dart game, there is no sequence for the places of the darts on the board [28]. CNN-based algorithms and YOLO versions are better solutions for our purpose.

CNN and YOLO are among the best and fastest algorithms for object detection [29, 30]. As Du mentioned [29], YOLO is derived from CNN and fastens up the object detection process.

All of the above works have powerful points about automatic dart scoring. However, they have deficits too. Commercialized products are expensive and not easily accessible. Gundjons' [19] and Hoettinger [18] approaches use a traditional model which is not as reliable as deep learning and the process is very slow. Martsyshyn's [17]project needs a completely different dart board with sensors which is not accessible to many people. McNally's work [16]is the most accessible and accurate one among the works we studied.

However, the success rate drops when different camera angles are used and the training process is slow. We offer a system which is accessible, faster and more accurate.

3. PROPOSED METHOD

3.1. Dart Game and Scoring

Before explaining our models, we can give some information about how a dart game is played and scored. A dart game is played between two individuals or two teams. There are many different dart games but the playing routine is the same with nearly all of them. The first player throws 3 darts to the board and gets his/her points according to the landing positions of the darts. Then, the second player throws his/her three darts. This process continues until one of the players/teams reaches the target.

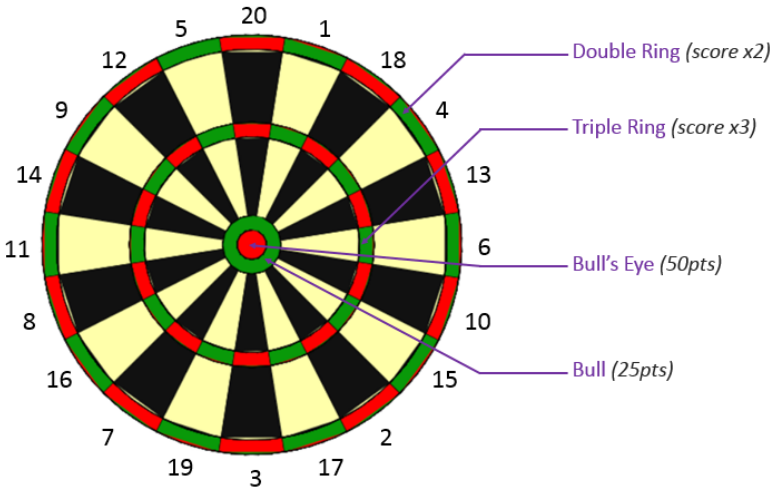

Figure 18. Bristol Style dart board and its regions

Figure 18 shows a Bristol Style dart board which is the standard one used in the tournaments. As can be seen, there are 62 different areas with different values. These areas are called single (exact value of the slice), double (single value x 2), triple (single value x 3), bull and bull's eye -or red bull- (bull x 2). When a dart hits the board, the points are calculated according to which part of the board is hit. Figure 19 shows a steel-tip dart. Although the shapes of the darts are very similar, they are not the same. Some players prefer to use a shorter or longer shaft or some prefer to use a thinner/thicker barrel.

Figure 19. A dart and names of its parts [33]

Popular Dart Games [1]: Among many game types, 2 of them are the most popular ones. Also, there are many tournaments for these two types of dart games.

The first one is “501” (the general name of this type is X01 because the number can be 301 or 1001 too. However, 501 is the main one which all official tournaments use): In this type of game, players start with 501 points and the one who reaches zero (exact zero), wins the game. However, the last dart should hit a double. For example, if the remaining point is 24, the player has to hit double 12. Here, the triple ring counts the triple value of the slice x 3 and the double ring counts the double value of the slice. As can be understood, the maximum value with 3 darts can be 180 (3 times triple 20). Bull counts as 25 and red bull – which is also a double – counts as 50 points.

Second popular game type is called “Cricket”. In a cricket game, players try to hit every predetermined point (generally 15 through 20 + bull) 3 times. For example, in a 15-20 cricket game, the first player who hits all numbers from 15 to 20 and the bull 3 times wins the game. Here, triple means 3 hits so hitting a number’s triple counts the same as hitting the number 3 times. Likewise, double counts as hitting the number 2 times.

Although the scoring systems are different among different types of games, the terminology is the same. If we know the position of the dart tip, we can make the calculations for scoring.

3.2. Our Method

Our purpose is to develop an automated dart scoring system which is easily accessible, cheap in price, fast enough to let the players play the game without any delay and can be used in homes and public places even in tournaments.

All the models mentioned in Related Works section (Section 2.2) have positive and negative sides. Commercialized systems generally use 3 cameras which makes them less accessible. Also, many of them are dependent on the board brands.

Gudjons' [19] and Hoettinger's [18] approaches have a low level of machine learning and their object detection process is very slow. Since there is no published study about their approaches, their success rate is not calculated. However, it is clear from their videos and the algorithms they used, the process is not as fast as desired.

Martsyshyn's approach [17]is very different according to other studies. However, their solution needs extra hardware and development which brings us to the same position with the commercialized products' low accessibility.

McNally's deep dart model [16]is developed to overcome these problems. As explained, there is no fixed model for dart games. Since the positions and the numbers of the darts thrown to the board are not certain (some darts may hit out of the scoring area), normal keypoint definition does not work. McNally et al. overcome this problem by defining keypoints as objects. By this way, they try to identify dart positions with one camera, independent of the camera angle. This approach is the closest one to our purpose. However, the Yolov4 algorithm they used, is now an old and slower one according to the newer versions. Furthermore, their idea to use a portable camera may be good to use in homes but it is not ideal for crowded places. Anyone, including players, can hit the camera and this can affect the whole system. Even if the camera is fixed, it has to be around the players. The danger is still present. Changed camera resolution, varying environmental lighting and camera quality are all parameters which can change the results. It can be understood from their tests with the second dataset. Their first dataset is created by using a fixed camera which is put near to the players. The success rate is 94.7%. Their second dataset is created with a portable camera and the shots are taken from different angles with different lighting. The success rate drops to 84%.

In our thesis, we combine all the good ideas of these studies. We use a single camera to make the system cheaper and eliminate calibration problems. For this single camera, we determined a fixed place which is not close to people. For the position of the camera, we decided to choose a place and angle very close to the view of the scorekeeper. It is nearly 45 degrees of angle to the board. We built a light ring to have homogenous light on the board to eliminate shadows and have a good view. We use a faster algorithm which is Yolov8.

First of all, we created a dataset. This dataset is a combination of 3 different datasets obtained from the internet. One of them is the dataset of “deep darts” [4]. This dataset has more than 10.000 images which are taken from a direct angle. Since their labelling is just for one point -intersection point of the dart and the board-, we used 350 of them just for test purposes. We got around 3.500 images as the main dataset from Roboflow [31, 32]. We augmented the images by fine-tuning color, brightness and angles. This made our dataset around 9.000 images. We re-arrange the labels and we fasten up the training and testing time by cropping out the unnecessary parts from the images that are outside of the scoring area (Figure 20).

Figure 20. Image cropping. The right image is the original image and the left is the cropped one.

To crop the images, we used Hough Circle Transform [34]. Cropping out the board area becomes easy by detecting the outer circle of the board with Hough Transformation. Here, cropping means making the unwanted area transparent and making its weight 0. With the help of this operation, we reduced the training time by 10%.

We aim to find every dart thrown to the board. While doing this, speed is as important as accuracy because dart is a game of concentration and no one wants to wait for scoring during the game. We need to use a fast deep-learning algorithm.

First, we tried to use RNN (Recurrent Neural Network) and especially LSTM (Long Short-Term Memory) to see the results. Obviously, LSTM -actually all RNN-based algorithms- is not the correct choice because LSTM needs to have a sequence of data to offer good accuracy. Its long-short memory is based on data which should follow a path. However, in a dart game, there is no ordered sequence. The past and future points are generally very different from each other. This means that LSTM is not a good choice for our project.

CNN (Convolutional Neural Network) seems one of the best solutions for our project. We tried R-CNN (Region-based Convolutional Neural Network) first. R-CNN had a 98%

success rate but it was slower than expected. To be faster, we tried Fast R-CNN. It was faster with nearly the same accuracy as R-CNN but still not fast enough. Faster R-CNN was the next algorithm to try. Again, the success rate was very good (97%) and the speed was very satisfying. As explained in Section 2.1, Faster R-CNN is a development over R-CNN and Fast R-CNN (Figure 21). Instead of selection search which R-CNN and Fast R-CNN used, Faster R-CNN uses a different network for regional predictions [35, 15].

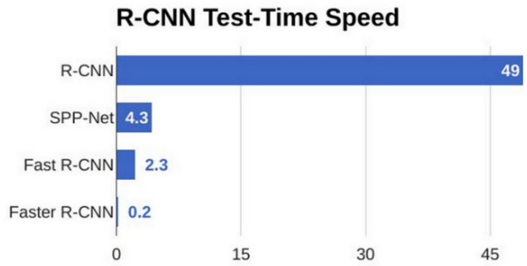

Figure 21. R-CNN speed tests

However, even Faster R-CNN was still slow and we wanted to try a faster and newer algorithm, which is YOLOv8.

YOLOv8 is the newest version of the YOLO series at the time of writing this paper. YOLOv8 is developed by Ultralytics [7] and their approach is different from many of the object detection algorithms, even from previous YOLO versions. YOLO (You Only Look Once), is developed by Joseph Redmon et al. [8], and the main aim of this algorithm is to detect objects very fast with high accuracy. The best part of YOLO is its one-stage object detection model. Many of the other CNN models use double-detection and this makes them slower.

Over time, many updates have been developed to the YOLO algorithm [9]. The difference of YOLOv8 is its anchorless architecture. Not having anchor boundaries, an algorithm can detect objects faster and these objects can easily be converted to keypoints [9].

In our model, the first step is to detect the darts. After testing our YOLOv8 algorithm with the dataset we prepared, we got 96.8% success. It is normal that the accuracy is lower than R-CNNs but YOLOv8 is much faster and the accuracy rate is acceptable. This speed/accuracy ratio is better than a more accurate but slower model for our purpose.

We were able to detect darts on the board. However, this is not enough for us. We need to find the tip of the dart and the position of the tip on the board. We calculate the score by using a score map on the board. To find an exact position, we need to know the coordinates and the angle to a reference point. Angle is necessary to find which slice is the dart in and coordinates are necessary to find which portion of the slice is the dart in (double, triple or single). Figure 22 shows the board map. Since there are 20 slices, the angle of every slice is:

$$R \div Sn = As \tag{3.1}$$

Here, R is the angle of a complete circle which is 360°. Sn is the total number of sections which is 20 and As is the angle of one slice, which is:

$$360 \div 20 = 13° \tag{3.2}$$

In order to determine the position, we need a reference point. In our model, our reference point is the center of the red bull ring. Again, by using Hough Circle Transform [34], we are able to detect the center circle which is the red bull circle. The center point of this circle is our reference point. Taking this point as the origin, we can draw x and y axes. By this way, it is possible to find the coordinate and the angle of the dart (Figure 22).

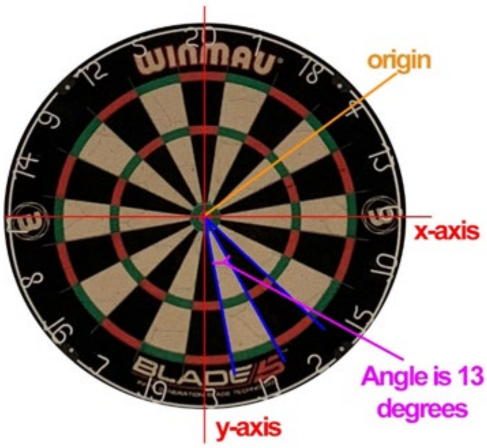

Figure 22. Dartboard axes

The most important thing in the model is to find the dart tip. For this purpose, McNally et al. [16], used a different technique which is not to find the darts but to find only the dart tips. They just labelled the tip part of the darts and trained the system with this dataset. This is a good idea with a good success rate with same-angle photos/videos. However, when the angle is changed, success rate drops dramatically (demonstrated with dataset 2). We think that finding the whole dart body is a better solution for accuracy with different angles. As mentioned in the experiments" section, we are able to detect the darts with a good success rate.

Figure 23. Training results

However, this is just the first part. After detection, objects -in our case: darts- are segmented in rectangles each called a "*bounding box*" [36] as done in all YOLO versions. It is not possible to find the tip of the dart within a rectangular shape. We needed to find a way to determine the position.

It is clear that the tip of the dart is on the shorter side of the rectangle. It can also be seen in Figure 23. The problem with this approach is not knowing the position of the tip on the side. As can be seen in the result photos, there might be 3 positions of the dart tip:

(1) It can be on the exact middle point on the side: If it is on the middle, this means that the dart is standing straight.

(2) It can be on one of the edges: This means the dart is standing at an angle and the projection of the tip is outside of the flight's boundaries.

(3) It can be on any point on the side: Sometimes, the dart stands at a very low angle, and the tip's projection is inside of the flight's boundaries. This time, the tip stands at some point on the side.

These standing positions change the area of the rectangle. If the area of the rectangle is below a certain amount, we can say that the tip is on the middle point. If the area is above a determined value, we should check the edges. Figure 24 shows the different rectangles. Red and blue rectangle covers angular darts and the green rectangle covers a straight dart. The area difference is obvious.

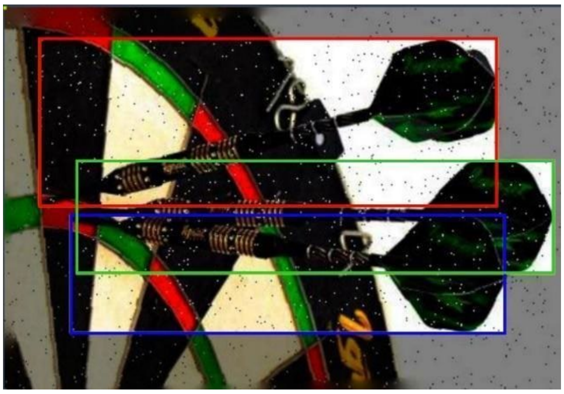

Figure 24. Different rectangular segments

3.2.1. Pre-calculations

Before talking about the solution proposals to find the tip, we can mention some pre-calculations which allow us to calculate the score without needing any other further operations. These are:

(A) As mentioned in position (1), if the dart is standing straight, the pin is on the middle of the short side. After finding the correct short side, we can find the position of the tip.

(B) If we know that the tip is on the edge (by calculating the rectangle area), after finding the correct edge, the score is calculated by using that edge as the dart tip.

(C) In some cases, we can calculate the score without having a necessity to find the tip position. If all the points on the intended side are only on the same segment, then there is no need to find the tip because all possible positions are on the same score area (see Figure 25).

Figure 25. Boundary types: For dart 1: The bottom side has the tip of the dart and all points of the side are on the same segment (single 18). Independent of the tip position, we can say that the score is 18. For dart 2: We can say that the tip is on the edge and both edge points are on singe 20. So, the score is 20.

These pre-calculations can save us a lot of time. However, for the cases except these 3, we need to find the dart tip.

To overcome this problem, we propose 3 different methods:

3.3. Proposed Methods

3.3.1. Barrel is Enough

As will be mentioned in the “Experiments” section (section 4), it is possible to detect a dart without its flight. Dataset can be arranged to be labeled from shaft to tip. When this is achieved, option (3) is eliminated because the width of the shaft is not that much bigger than the tip so the projection of the tip cannot be inside the shaft’s boundaries. If this is the case, we can say that the tip is either on the middle point or on one of the edges. So, the score can be calculated with method (A) or (B) in the pre-calculations section. This is a good method for angled views. However, with a direct view, it may be very hard to detect a dart because the flight blocks the view and the barrel cannot be seen.

3.3.2. DartEasy Trio

Barrel detection method is an effective method when the angle allows us to see the shaft and the barrel clearly. When the view angle changes and does not let us detect the shaft and the barrel, we need to detect the dart as a whole. This time, flight should also be in the calculations.

If we connect the outline points of a dart, the shape is very close to a triangle (see Figure 26). This triangle shape is used in Gudjons’ approach [19]. Their computational way was very different and had deficits which have been written here. With our approach, we detect the dart with deep learning and we propose an effective method to find the tip of the dart. The rest is doing some calculations for computing the score.

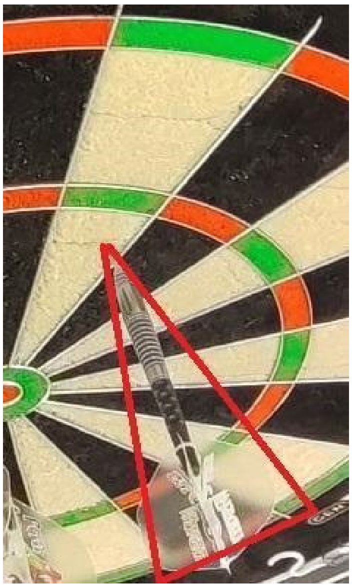

Figure 26. Triangular shape of a dart.

One of the edges of the triangle, which is the edge created by the longer sides of the triangle, should be on one of the short sides. The triangle should intersect with the rectangle with at least 3 points.

YOLO's bounding boxes have 5 parameters. These are: "label, x, y, w, h". By using these parameters, we can find any point on the bounding box. If we trace the short side of the rectangle from one edge to another and use the other short side as the side of the triangle, we find the tip at one point.

$$y = ax + b \tag{3.3}$$

Above equation is the equation of the line. We need to travel across this equation to search every point. To find which point is our point, we need to use segmentation. After having the segmentation, we can calculate the intersection of the segmented area and the triangle area. The point which gives us the biggest intersection, is our point (Figure 27).

$$IOU = \text{Area of Intersection} \div \text{Area of Union} \quad [37] \tag{3.4}$$

Here, IOU is Intersection Over Union. Area of Intersection is the intersected part of our triangle with the segmented object. Area of Union is the total area of the segmented object and our triangle. IOU_{max} is the value we are looking for.

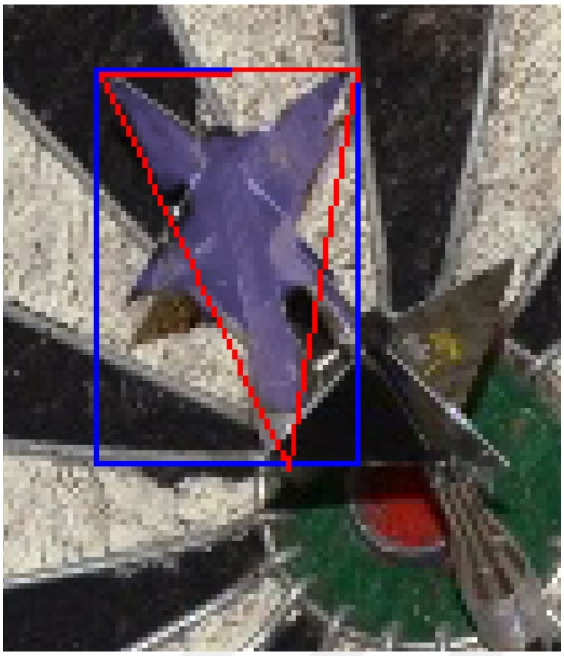

Figure 27. Bounding box (blue), Model rectangle (red) and segmented area (light blue)

This approach gives us an opportunity to detect some darts which cannot be seen clearly. We think that the accuracy rating will be very high according to all of the models presented. However, the necessity of segmentation is a retarding process which causes us to lose time.

3.3.3. DartEasy OBB

Oriented Bounding Box (OBB), is a YOLO feature which YOLOv8 supports [36]. With the OBB, YOLO can define bounding boxes with angles and the angle is related with the object (Figure 28). Unlike the normal bounding box which has *x-center, y-center, width and height* properties, the four corner points define YOLO OBB and the coordinates of these points are normalized. This is the format: $BB = (i, x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4)$ in which i denotes the class_index [36]

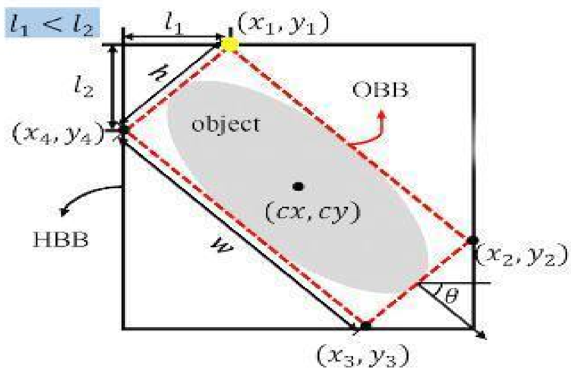

Figure 28. Showing OBBs for an elliptical shape. [36]

There are three types of OBB:

- ϴ-based OBB: this type uses the values of the parameters (cx, cy, h, w, ϴ)
- Point-based OBB: The parameters used in this type are: x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4
- h-based OBB: This type uses the parameters: x_1, y_1, x_2, y_2, h

As can be seen in Figure 28, YOLO can turn the bounding boxes according to the object. If this can be achieved correctly, thanks to the symmetrical shape of the dart, the tip of the dart always be on the middle point of one of the short sides. To find the position easily, "Point-based OBB" seems the best option. For example, if the dart tip is on the (x3, y3), (x2, y2) line on Figure 28, the coordinates of the tip is:

$$x_{ip} = (x_3 + x_2)/2,\ y_{tip} = (y_3 + y_2)/2 \tag{3.5}$$

However, in order to use the OBB method, the dataset needs to be created compatible with OBB. Our dataset was prepared in a traditional way, so we cannot provide results for this method in this thesis.

3.4 Algorithm

To detect the dart fast, we use the same mentality as the "binary search" technique in data processing [38]. Binary search algorithm provides us with a simple and fast search method for sorted data. Since our coordinates are sorted, we can modify and use this algorithm. In binary search, we first look at the middle of the stack, we compare the value in the middle with the first and last value. By this comparison, we can understand which side includes our search point. If it is in the last half, we accept the middle point as the new start point and forget about the first half of the data stack. If the wanted point is in the first half of the stack, we accept the middle point as the new endpoint and do not think about the last half of the stack anymore. This process repeats until we find the intended point.

To find the dart tip, we use a binary search-like algorithm. We first check the edges and the middle (Figure 29). For "Barrel is Enough" (section 3.3.1) and "DartEasy OBB" (3.3.3) methods, this is enough and we find the tip because, as explained in the related sections, the tip is on one of these points. By this method, even if it is not on one of these points, we can understand which half of the side has the tip from the area of the segmentation and our triangle. After finding which half we should look at, we accept the middle point as the new start or end point according to the half we are looking for. We can repeat this process until we find the tip.

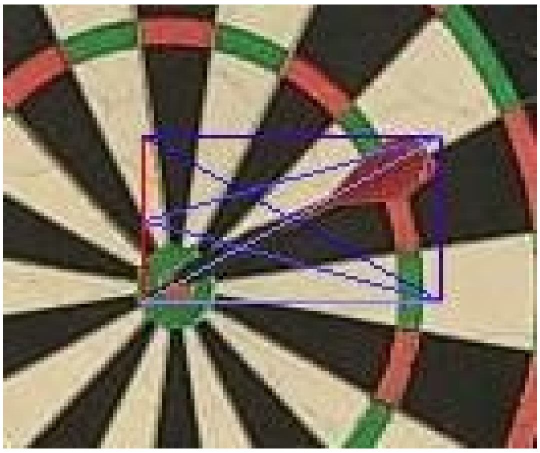

Figure 29. First 3 triangles we draw into the bounding box.

After finding the tip, we need to find the region that the tip hit. We know that we have 6 circular regions on the dartboard. These regions are (starting from the outer border): double ring, single ring, triple ring, single ring again, bull and red bull. To proceed, we need to draw an imaginary line from the center (reference point) to the tip. According to the distance of the tip from the reference point -centre point- (the length of the imaginary line), we can determine which region the tip is in. As the last step, we find the angle between the newly created line and x-axis to find which point is our point.

In order to find the circular regions on the dartboard, we again use "Hough Transformation" [34]. The problem we might face with this approach is the changeable circle diameters and distance parameters due to the distortion because of the camera angle. So, we need to first find the proportions of the distances to the centre point. Since we know the original diameters and lengths, we can calculate the new proportions. For this purpose, we should first find the distance from the centre to the top point on the outer circle.

4. EXPERIMENTS

During our experiments, we evaluated different datasets and methods compared with the state-of-art studies in the literature. The datasets we used and the results will be presented in this section.

The test results are all measured with an overlapping system, which is the intersection area of the labelled data with the detected object's area [39, 40].

We used 3 different datasets during our experiments. First one is the Deep Darts [16] dataset (Figure 30). This is to evaluate our system's performance. How cropping affects, how YOLOv8 differs etc. Our tests were satisfying. Deep-dart method is successful when they use dataset 1 (fixed camera position, fixed angle). However, the success rate has dropped when they use dataset 2 (different angles). We tried to achieve better and faster results.

Table 1. Comparison between DeepDarts dataset 2 and DartEasy. Scores are calculated with "overlapping"

	DeepDarts	**DartEasy**
Training Time	7min/epoch	5min/epoch
Test Score	84.0%	85,2%

The speed difference is related to both image cropping and the method. Cropping images made a around 10% difference. YOLOv8 made a real difference. We, then, used some of the images in this dataset with our final dataset.

Figure 30. DeepDarts dataset samples. Images in the first row are from dataset1 and images in the second row are from dataset 2

Second dataset we used is "*Dart_detection_v2*" by Markowich, taken from *Roboflow* (Figure 31) [31]. This dataset contains 1,086 images in total. Markowich labelled only the shaft + barrel parts of the darts. We used this dataset for our "Barrel is Enough" model (section 3.3.1). This dataset with our model achieved great success. With our first setup, which is a fixed camera angle on the light ring, we got a 99.91% success rate with overlapping features. One of the advantages of this dataset is not needing the flight. So, even if the flight falls outside of the camera view, we could detect the darts. This helps to put the camera to narrower places (like the light ring). However, when the angle becomes more direct -for example looking at the board from a direct angle- success rate drops dramatically. One of the reasons for this drop is the blocked view. Because of the flight, the barrel and the shaft have less presence in the image. This makes the model skip some darts in the images.

Figure 31. Dart_detection_v2 dataset. Both raw images and labelled images can be seen.

Third dataset is the "*Dart Object Detection Dataset*" taken from Roboflow (Figure 32) [32]. This dataset has 4,398 images and contains different types of darts with very different backgrounds and positions. The author used many images from many different sources like advertisements, dart games, product photographs etc. So, there are darts in very different angles both with and without a board. Also, there are images with environmental backgrounds like people, dart accessories etc. This makes us detect darts with very different angles on the board. This dataset is nearly perfect for our "DartEasy Trio" and "DartEasy OBB" models (sections 3.3.2 and 3.3.3). However, since there are too many different kinds of images, in order to get good results, the epoch number should be high. We got a success rate over 99% after 100 epochs with a fixed-angle camera. Variety of the images is good but there are some images which confuse the model and make training harder. With some modifications, this dataset may work better.

Figure 32. Dart Object Detection Dataset. Both raw images and labelled images can be seen

At last, we combined these 3 datasets. Our aim was to detect both flightless and complete darts with varying angles. We achieved the expected results with some test images but could not get the desired results with others. We think that the reason for this is the labelling differences between datasets. For example, the first dataset labelled the images without flights and the other labelled with flights. This caused the model to detect more than one result with a single dart. We think that with some adjustments and modifications to the dataset may give us perfect results. Figure 33 and 34 shows some results of our tests.

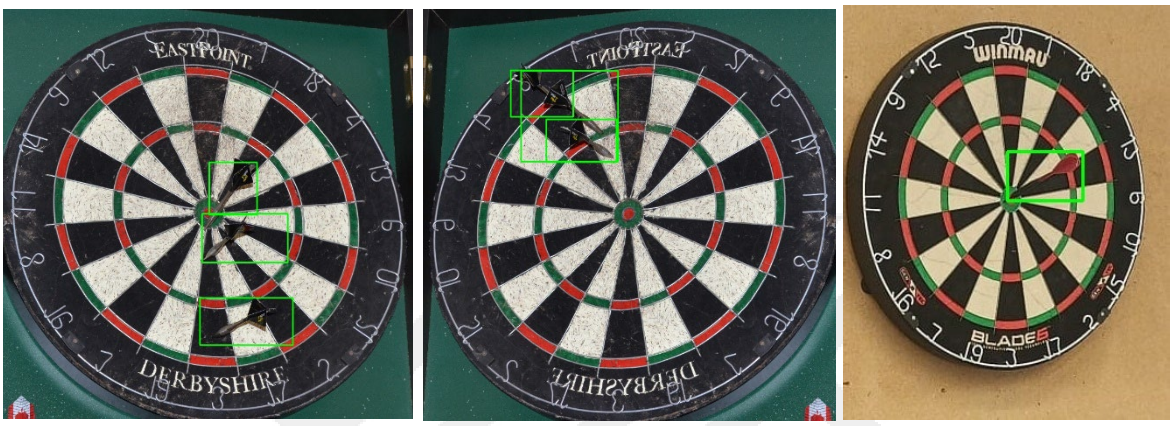

Figure 33. Some results of our DartEasy algorithm

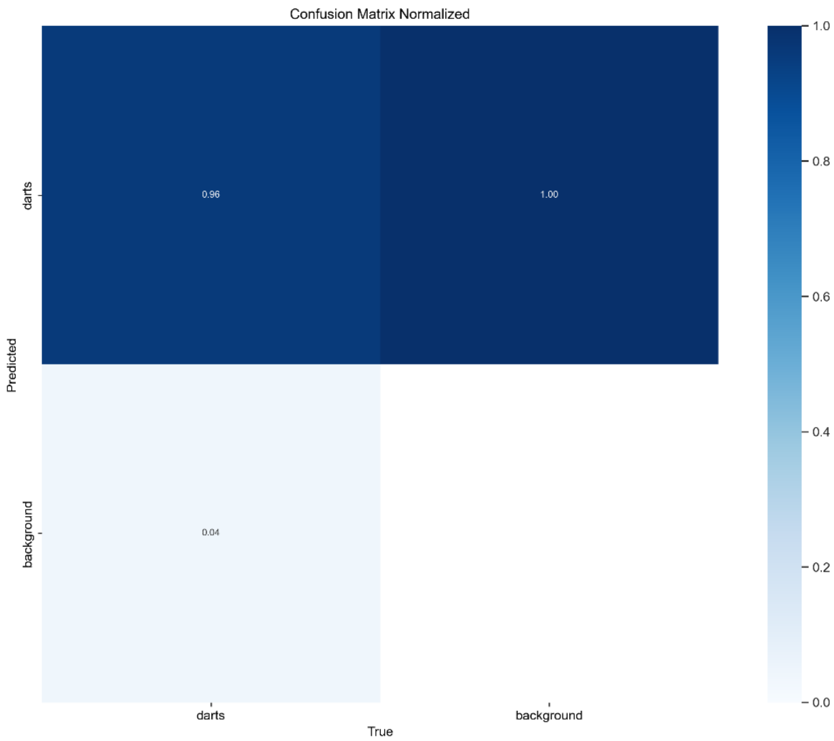

Figure 34. Confusion matrix of a DartEasy training with 100 epochs

As mentioned above, cropping unnecessary regions gained us time and accuracy. However, during our experiments, we saw that cropping with Hough Circle Transform is not efficient with every image. If there is a camera angle below 70 degrees or above 110 degrees, Hough Circle Transform cannot get the board dimensions perfectly because of the distortion in the image. For example, with an image such as shown in Figure 35, the detected circles do not cover the board perfectly. We can use these circles for cropping but that is not the ideal case and if the angle gets lower, Hough Circle can crop some necessary points too (Figure 36). This shows us that this cropping model is not useful for every angle. It can be seen in Figure 35 that there is no problem with detecting the red bull ring. We can still find the centre point of the board with the same method. This is because the red bull circle is so small that the distortion does not affect it much.

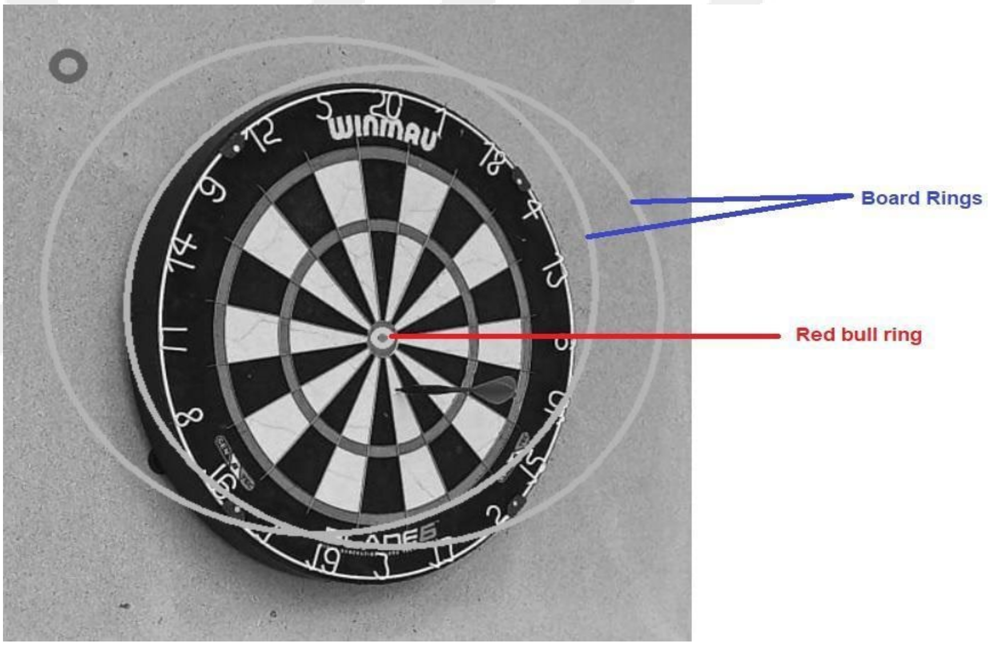

Figure 35. Red bull ring found perfectly. However, Hough Circle Transform couldn't detect the board shape because of the distortion. Both circles can be used for cropping since they include all valuable data but they are not ideal.

Figure 36. Cropped versions of Figure 33 for both of the found circles. Each circle is cropped separately.

In order to improve this result and get perfect crops every time, a mapping can be applied to the image. This mapping is to get the dartboard and make it face directly. If one can achieve that, cropping would always fit perfectly. The implementation of this idea is reserved for future studies.

5. CONCLUSION

In this thesis, we have offered an object detection-based algorithm to calculate dart scoring automatically which can be used everywhere including homes and public places. We proposed several different methods to detect darts on a dart board to calculate the score automatically with deep learning. We used pre-built datasets for this thesis and tried different approaches to find the best solution. Furthermore, we proposed three different methods and achieved very good results within some restricted conditions. For this purpose, we first put one camera with a fixed angle. Our results are very satisfying with this setup. We also changed the setup and tried different camera angles. This time the success rates dropped but still we got satisfying results. Again, these tests showed us better results can be achieved with some modifications. We are very satisfied with the speed of YOLOv8 and convinced that this speed is more than enough for a dart game. As a future study, we are planning to create our own dataset and combine all three models to get perfect results. Furthermore, a different algorithm can be used for cropping. As explained in the experiments, the positive effect of cropping is obvious but our cropping method is not feasible for every angle for the camera. One camera model is a very good solution for many cases but when the darts are very close, one camera cannot see the second dart because of the occlusion of another dart. For better results, a 2-camera system can also be examined as a future study. We hope this thesis can help to have more joyful dart games.

REFERENCES

[1] Darts | Rules, History & Equipment. (1998, July 20). Encyclopedia Britannica. https://www.britannica.com/topic/darts

[2] DartsWDF. (2024, January 10). DartsWDF. https://dartswdf.com

[3] S. Albawi, T. A. Mohammed and S. Al-Zawi, "Understanding of a convolutional neural network," 2017 International Conference on Engineering and Technology (ICET), Antalya, Turkey, 2017, pp. 1-6, doi: 10.1109/ICEngTechnol.2017.8308186

[4] R. Yamashita, M. Nishio, R. K. G. Do, & K. Togashi, “Convolutional neural networks: an overview and application in radiology”, 2018, Insights into imaging, 9, 611-629.

[5] A. John, & D. Meva, “A comparative study of various object detection algorithms and performance analysis,” 2020, International Journal of Computer Sciences and Engineering, 8(10), 158-163.

[6] N. Yadav, & U. Binay, “Comparative study of object detection algorithms,” 2017, International Research Journal of Engineering and Technology (IRJET), 4(11), 586-591.

[7] Ultralytics | Revolutionizing the World of Vision AI. (n.d.). https://www.ultralytics.com

[8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” In Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 779-788

[9] J. Terven, D.M. Córdova-Esparza, and J.A. Romero-González, “A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS,” Machine Learning and Knowledge Extraction, 2023, 5(4), 1680-1716.

[10] S. Liu, H. Zhou, C. Li and S. Wang, "Analysis of Anchor-Based and Anchor-Free Object Detection Methods Based on Deep Learning," 2020 IEEE International Conference on Mechatronics and Automation (ICMA), Beijing, China, 2020, pp. 1058-1065, doi: 10.1109/ICMA49215.2020.9233610.

[11] T. Cheng, L. Song, Y. Ge, W. Liu, X. Wang, & Y. Shan, “YOLO-World: Real-Time Open-Vocabulary Object Detection,” 2024, arXiv preprint arXiv:2401.17270.

[12] F. Joiya, “Object detection: Yolo Vs Faster R-Cnn,” 2022, Int Res J Modern Eng Technol Sci, 9, 1911-1915.

[13] C. Wang, Z. Luo, S. Lian, S. Li, “Anchor Free Network for Multi-Scale Face Detection,” 2018, 1554-1559. 10.1109/ICPR.2018.8545814.

[14] S. Fu, Y. He, X. Du, “Anchor-free object detection in remote sensing images using a variable receptive field network,” 2023, EURASIP J. Adv. Signal Process. 2023, 53, https://doi.org/10.1186/s13634-023-01013-2

[15] Gandhi, R. (2018, December 3). R-CNN, Fast R-CNN, Faster R-CNN, YOLO — Object Detection Algorithms. Medium. https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e

[16] W. McNally, P. Walters, K. Vats, A. Wong and J. McPhee, “DeepDarts: Modeling Keypoints as Objects for Automatic Scorekeeping in Darts using a Single Camera,” IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2021, pp. 4547-4556

[17] R. Martsyshyn, Y. Miyushkovych, L. Sikora, N. Lysa, R. Tkachuk “Technology of Remote Recognition the Dart-Arrow on the Target,” IEEE Second International Conference on Data Stream Mining & Processing, 2018, 10.1109/DSMP.2018.8478618

[18] H. (n.d.). GitHub - hanneshoettinger/opencv-steel-darts: Automatic scoring system for steel darts using OpenCV, a Raspberry Pi 3 Model B and two webcams. GitHub. https://github.com/hanneshoettinger/opencv-steel-darts

[19] L. (n.d.). GitHub - LarsG21/Darts_Project: This Project is used for automatic dart scoring using computer vision/. GitHub. https://github.com/LarsG21/Darts_Project

[20] Automatic dart scoring system | SCOLIA. (n.d.). Scolia. https://scoliadarts.com/

[21] Autodarts.io | Automatic darts scoring system. (n.d.). https://autodarts.io

[22] Dartsee - Automatic Darts Scoring | Social Gaming Experience. (n.d.). Dartsee. https://www.dartsee.com

[23] Dartsee - Automatic Darts Scoring | Social Gaming Experience. (n.d.). Dartsee. https://www.dartsee.com

[24] S. C. Hsia, S. H. Wang, W. C. Cheng, & C. Y. Chang, “Intelligent Blowgun Game Scoring Recognition System Based on Computer Vision,” 2021, IEEE Access, 9, 73703-73712

[25] T. T. Zin, I. Oka, T. Sasayama, S. Ata, H. Watanabe, & H. Sasano, “Image processing approach to automatic scoring system for archery targets,” 2013, Ninth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (pp. 259-262). IEEE.

[26] R. K. Mishra, G. Y. Sandesh Reddy, Himanshu Pathak, "The Understanding of Deep Learning: A Comprehensive Review", 2021, Mathematical Problems in Engineering, vol. 2021, Article ID 5548884

[27] Y. LeCun, Y. Bengio, & G. Hinton, “Deep learning,” 2015, Nature, 521(7553), 436-444.

[28] A. Sherstinsky, “Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network,” 2020, Physica D: Nonlinear Phenomena, 404, 132306.

[29] J. Du, “Understanding of object detection based on CNN family and YOLO,” 2018, In Journal of Physics: Conference Series (Vol. 1004, p. 012029). IOP Publishing.

[30] C. Liu, Y. Tao, J. Liang, K. Li and Y. Chen, "Object Detection Based on YOLO Network," 2018 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, China, 2018, pp. 799-803, doi: 10.1109/ITOEC.2018.8740604.

[31] Dart_detection_v2 Instance Segmentation Dataset (v1, 2023-01-25 12:12pm) by Marcowich. (n.d.). Roboflow. https://universe.roboflow.com/marcowich/dart_detection_v2-hr0ds/dataset/1

[32] Dart Object Detection Dataset (v3, Dart_2022-05-13) by MediumLevel. (n.d.). Roboflow. https://universe.roboflow.com/mediumlevel/dart-nacyr/dataset/3

[33] Michael. (2021, November 3). Dart parts: The anatomy of a dart. Darts Dojo. https://dartsdojo.com/dart-parts/

[34] D. J. Kerbyson, & T. J. Atherton, “Circle detection using Hough transform filters,” 1995.

[35] S. Ren, K. He, R. Girshick, J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” Advances in neural information processing systems, 2015, 28

[36] U. (n.d.). Oriented Bounding Box (OBB) Datasets Overview. Ultralytics YOLOv8 Docs. https://docs.ultralytics.com/datasets/obb/?query=bounding+box

[37] M. A. Rahman, Y. Wang, “Optimizing intersection-over-union in deep neural networks for image segmentation,” In International Symposium on visual computing, 2016, pp. 234-244

[38] T. Seidl, J. Enderle, “Binary Search In: Vöcking, B., et al. Algorithms Unplugged”, (2011), Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-15328-0_1

[39] C. Liu, Y. Ren, M. Liang, Z. Gu, J. Wang, L. Pan, Z. Wang, "Detecting Overlapping Data in System Logs Based on Ensemble Learning Method", 2020, Wireless Communications and Mobile Computing, Article ID 8853971

[40] R. Gontijo-Lopes, Y. Dauphin, & E. D. Cubuk, “No one representation to rule them all: Overlapping features of training methods,” 2021, arXiv:2110.12899.