HoloNav: HoloLens as a Surgical Navigation System

Detecting optical reflective spheres using YOLOv5 and the Hololens' grayscale cameras

Ee Xuan Tan
Supervisor(s): Ricardo Guerra Marroquim, Mohamed Benmahdjoub, Pierre Ambrosini
EEMCS, Delft University of Technology, The Netherlands

A Dissertation Submitted to EEMCS faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 19th, 2022

Abstract

Surgical navigation is a tool that surgeons rely on everyday to perform accurate surgeries all over the world. However, this technology requires good hand-eye coordination and a high level of concentration. HoloNav is a project that inquires to see if using the HoloLens and augmented reality can replace the current surgical navigation methods. To do so, the HoloLens must be able to identify the patient and the location of the surgery instruments, which uses optical reflective spheres. This study focuses on using the grayscale cameras of the HoloLens and a deep learning algorithm YOLOv5 to test if it is possible to precisely detect optical reflective spheres. 3 models were trained with two different data sets, where the results show that the model trained on a data set would perform well on the validation set. However, they would perform far worse when exposed to a data set it was not trained on.

1 Introduction

During surgery, surgeons make use of tools and technology that guide them throughout the procedure. Surgical navigation helps the surgeon to locate the position of surgical instruments with respect to the patient's anatomy, thus answering the following questions, for example; Where is the targeted tumour? How do I reach it safely and where is the surgical tool currently at? [9]. Generally, optical surgical navigation uses a stereoscopic infrared camera, a screen and the navigation software. There are many benefits to surgical navigation in neurosurgery; a few examples such as supporting minimally invasive surgeries, increased confidence and preservation of the neurological functions of a patient. Furthermore, surgical navigation is used in orthopedic surgery to get a higher accuracy when positioning the implants [9]. However, current navigation systems visualize information on a 2D screen to which a surgeon need to look and switch attention between the screen and the surgical site. This requires additional coordination from the surgeons.

A suggestion to tackle this problem is using augmented reality (AR), which projects virtual objects over physical objects. According to a study conducted by the Erasmus MC, AR shows potential clinical feasibility of the Hololens 2 (HL2) for brain tumor surgery [4]. In order to do so, it is necessary to find the position of the tools relative to the patient. The research on HoloNav focuses on identifying the feasibility and accuracy of using AR to replace traditional surgical navigation tools. The eventual goal is to build the functionality necessary to use the HL2 to perform surgeries.

This work will primarily focus on the following question: "Can we use the object detection algorithm YOLOv5 to precisely detect the optical reflective spheres using the HL2 gray scale cameras?". YOLO is an acronym for the existing object detection algorithm, 'You Only Look Once' [13]. YOLO detects objects using a convolutional neural network and requires only a single propagation forward through the network. The key advantages of YOLO are that it is able to detect images extremely fast, with high accuracy and has strong learning capabilities. YOLO has gone through many iterations and improved in both speed and accuracy. In this work, the algorithm that is used to detect optical spheres will be YOLOv5. YOLOv5 has a series of different models that consist of different number of layers in the network, where there is a trade-off between speed and accuracy.

This paper is structured as follows; the related work can be found in Section 2. The methodology of how this work was conducted can be found in section 3 of this paper. Following, in Section 4, the results are discussed and the different trained models compared. In Section 5, the results, the contribution towards HoloNav and the ethical aspects of this research are discussed. Conclusions and the future work of the conducted work can be found in Section 6.

2 Related work

Object detection allows for the localization of objects of interest in an image. Object detection differs from algorithms that use standard convolutional networks in the way that the output can vary from image to image as the number of objects is not constant. This means that the length of the output layer varies. For this, other algorithms have been explored such as Faster R-CNN [14] and Single Shot Multi detector (SSD) [8]. According to this study on real-time vehicle type recognition [7], where YOLOv4 was compared to these state-of-the-art object detectors, it outperforms both detectors in terms of speed and accuracy. YOLOv4 runs as high as 45 frame per seconds (FPS) [2]. Now, if YOLOv4 is compared to YOLOv5, the object detector YOLOv5 can run inference up to 140 FPS with similar performance in accuracy to that of YOLOv4, according to this Roboflow blog post [10]. However, studies such as "Autonomous Vision-Based Primary Distribution Systems Porcelain Insulators Inspection Using UAVs" [12], show results that YOLOv4 outperforms YOLOv5 in terms of accuracy. This means that the literature is yet to reach a consensus on whether YOLOv4 or YOLOv5 is better. The varying results from the two models are related to different factors such as the hyper-parameters and the data set used [11]. However, since reflective spheres are not heavily detailed, it is believed that a smaller model of YOLOv5 could be used to train and run inference at a high speed. Therefore, YOLOv5 was the chosen object detection algorithm for this study. Although, for future, it would be good to verify the actual performance between YOLOv4 and YOLOv5.

3 Methodology

There are three sub-questions that need to be answered before being able to use YOLOv5 to detect the optical reflective spheres of the HL2 grayscale cameras. These consist of:

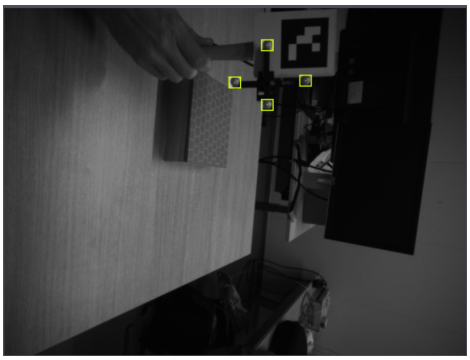
- How can we create bounding boxes on the grayscale images from the large data sets consisting of 1000+ images?
- How do we generate the text files to feed YOLOv5 the correct data?

- How accurate does YOLOv5 perform against the data sets? What are the limitations?

3.1 Bounding boxes

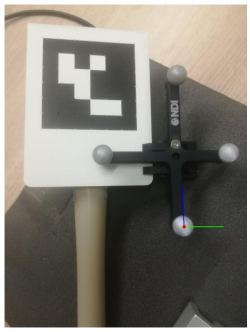
Generating bounding boxes for the data set images can be done through two methods. The first is to manually annotate the images in Roboflow [16], a computer vision developer tool to help preprocessing and model training. Roboflow can be used to draw bounding boxes manually and annotate the object. This ensures accuracy, however, is not scalable with large data sets.

Figure 1: Bounding boxes drawn on example image from data set



Alternatively, it is possible to automatically create annotations for the bounding boxes by using the center of the spheres and estimating the boxes using a linear regression approach. This is based on the largest and smallest spheres in the data set. The center of the spheres can be automatically calculated using the infrared cameras, which can identify the position of the reflective sphere. Using a QR code which the HL2 can detect, and a pointer detected by the infrared cameras, a calibration matrix can be computed between spheres and the attached QR code (see Figure 2) [1].

Figure 2: QR code and optical reflective spheres



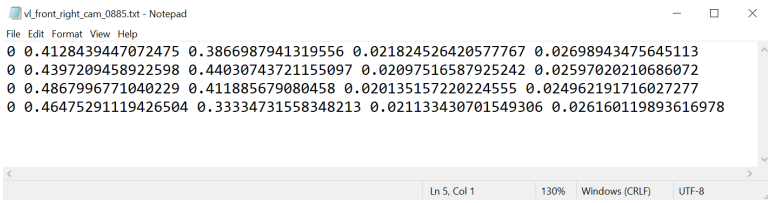
In this way, the positions of the reflective spheres can be found in the HL2 camera's coordinate system and the 3d coordinate points can be found. Now, by manually drawing the bounding boxes of the furthest sphere and the closest sphere, the bounding boxes can be drawn for all spheres based on its relative position in the camera coordinate space. The furthest sphere that is found corresponds to the smallest possible bounding box, while the closest sphere will have the largest bounding box. The bounding boxes of the other spheres are automatically calculated based on the distance in the coordinate space.

3.2 Obtaining the correct data for YOLOv5

YOLOv5 takes images and labels as inputs for its training. The labels are recorded in a text file, which is formatted as follows:

[Class number] [Bounding box center x location] [Bounding box center y location] [Bounding box width length] [Bounding box height length]

Figure 3: Example label file with 4 bounding boxes



Since YOLOv5 is a multi-object detector, the class number refers to the type (class) of object that is annotated by a bounding box. The bounding box center locations refer to the respective X and Y coordinates of the center of the bounding box. While, the width and height of the bounding boxes are referred to as how big the dimensions should be, these dimensions and the center of the bounding boxes are normalized.

3.3 Measuring results of YOLOv5

To train and assess YOLOv5's performance, the model needs to be trained, validated, then tested. It is important to split the data so that it gives the model sufficient images to train on, validate, and then finally test the model's performance. Following the training of YOLOv5, the weights can be used to run inferences on the test set.

Following the inference, the model will detect objects with bounding boxes; these boxes may differ from the annotated boxes. A corrected detected sphere is a true positive (TP). A detected sphere that is not actually a sphere is a false positive (FP) and a sphere that is not detected at all is a false negative (FN). These measures can be used to see how well the trained model performs. Furthermore, a common metric that is used to measure the performance of a model is the mean average precision (mAP); this is a measure of the average precision values calculated over recall values from 0 to 1. It takes the area under the precision recall curve, giving a good idea of the overall accuracy of the model. The mAP can be denoted as:

$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i$$

An mAP is determined based on its intersection over union (IoU) threshold, this is the area of overlap area of union. The

area of overlap refers to the area where the predicted box and the ground truth box overlap. While, area of union refers to the area of predicted box, area of ground truth box and area of both. A prediction that has a higher IoU than the threshold is a TP. [3] The mAP uses the precision and recall, where the precision refers to the number of positive predictions that were predicted correctly and can be denoted as:

$$Precision = \frac{TP}{TP + FP}$$

The recall on the other hand refers to the number of positives that were identified correctly which is written as:

$$Recall = \frac{TP}{TP + FN}$$

Another metric to check the models precision is to compare the center of the annotated bounding boxes to the bounding boxes of the predicted ones. This is measured in the number of pixels. This value can give an idea of how accurately the model is able to detect spheres. The average number of pixels can be calculated by

$$\text{Average pixel distance} = \frac{1}{N}\sum_{i=1}^{N} a_i$$

where for each sphere, i, the distance difference between the centers is denoted by a. Given the average, it is important to observes the variability of the data. The variance and standard deviation can show the spread of the data. These can be calculated as:

$$\text{Standard Deviation} = \sqrt{\frac{\sum(x_i - \mu)^2}{N}}$$

$$\text{Variance} = \sqrt{\frac{\sum(x_i - \bar{x})^2}{N - 1}}$$

4 Results

Two different image data sets were used to train the YOLOv5s model and measure its performance. The first set, data set A is the initial set that was a collection of images from the left and right gray scale cameras of the HL2. Data set A consists of 1023 images. An example of an image from data set A can be seen in Figure 4. These images have a consistent background, with little noise.
The second data set, data set B, was predicted to challenge the model more due to more elements and noise in the background. An example image of this data set can be seen in Figure 5. Data set B consists of 1392 images of which the optical spheres are placed at various distances and locations.

Figure 4: Example image of data set A

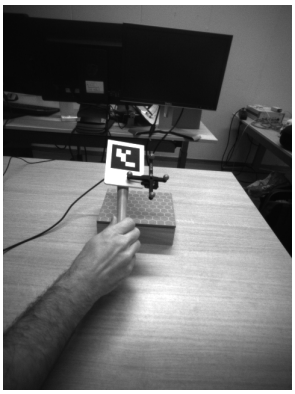
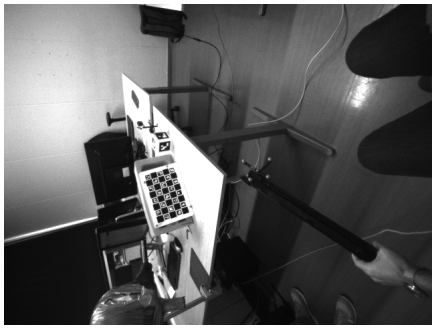


Figure 5: Example image of data set B



4.1 Training on data set A

Data set A was split as follows; 715 images were used to train, 204 images were used to validate, and the remaining 104 images were used to test. This was split following a 70% for training, 20% validation, and 10% for testing given the size of data set A [5]. Furthermore, the training data was trained using 150 epochs with the YOLOv5s model, which is the smallest and fastest model available according to the Ultralytics Github repository, where YOLOv5 is released [6]. Here below the results of the training will be discussed. The loss function of YOLO consists of 3 parts, training box loss, training objectness loss and classification loss. The box loss represents the error of the predicted bounding box in comparison to the annotated box. The objectness loss is the confidence of an object being in that bounding box. Lastly, the classification loss refers to the misclassification of the objects [13]. In Figures 6 and 7, the training box loss and the training objectness loss can be found. Both of these graphs show that the model is learning to detect the spheres and is able to converge towards 0. As for the classification loss, this is constantly 0 because optical spheres are the only class that the model is predicting; therefore there is never a misclassification.

Figure 6: Training box loss

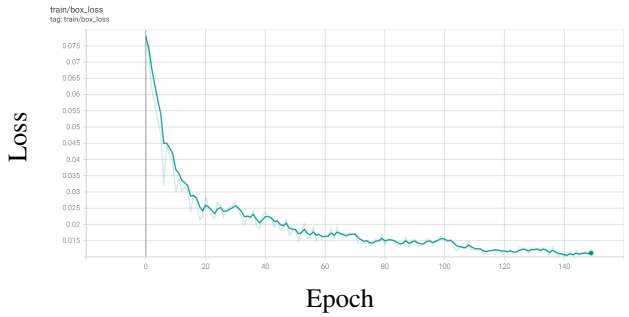
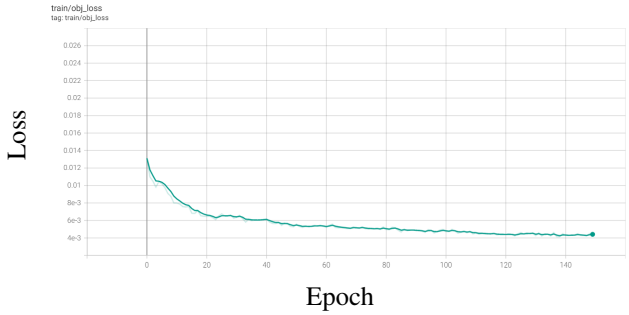


Figure 7: Training objectness loss



Furthermore, the models precision, recall and mAP can be found in the Figures 8, 9 and 10 respectively. These graphs indicate that precision, recall and mAP@0.5 converge to 1 and that the model learns to do this quite quickly. This result is a good indication that the model is able to learn to detect optical spheres fast and accurately. However, a big limitation here is that the data set may only be learning to detect optical spheres in simple images with little noise. The trained model was tested against two different sets, the first being the validation set of data set A and secondly, the entire set of data set B. The reason for this was to see first if it is able to learn to detect optical spheres in data set A and next to that see the performance of this model given a completely new data set. The results of the percentage of FT, FP and FN can be found in Table 1; the model performs well against the validation set of data set A. Here, there were only 5 FP and 7 FN with 809 being correctly detected. The mAP@0.5 is almost perfect with a score of 0.9945. Looking at the results, it is evident that the model performs well when trained on set A. However, when testing it against set B, the results are not as strong, with the model only detecting 2551 TP out of a total of 5568 annotated bounding boxes. Not only does it not detect all the bounding boxes, but it detects 2837 FP and 3017 FN bounding boxes. This results in a low mAP@0.5 score of 0.3629.

Table 1: Trained model on data set A: TP, FP, FN and mAP@0.5

Data set	**TP**	**FP**	**FN**	**mAP@0.5**
A: validation set	809	5	7	0.9945
B: entire data set	2551	2837	3017	0.3629

Figure 8: Training precision

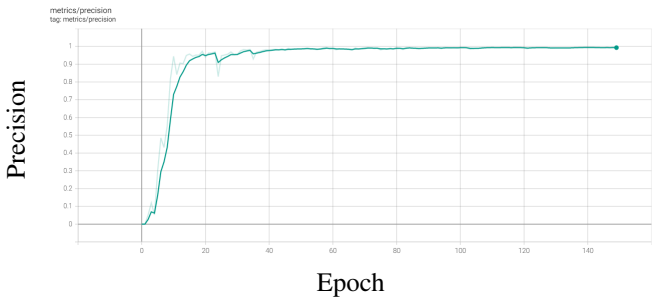


Figure 9: Training recall

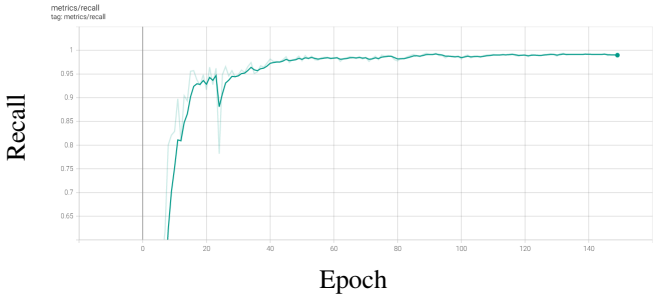
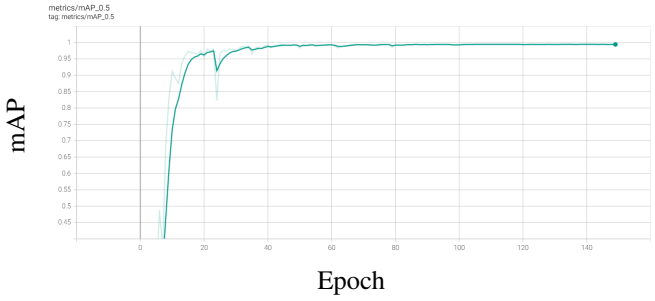


Figure 10: Training mean average precision@0.5



The average pixel distance of the trained model can be found in Table 2. Here, the average difference between an annotated distance and the predicted box is 1 pixel with a low variance and standard deviation of 0.4165 and 0.5589, respectively. The trained model is able to perform very well on validation set A. While, on data set B, it is evident that the model performs significantly worse with an average pixel distance of 9.0726 with a high variance and standard deviation.

Table 2: Pixel distance of model trained on set A

Data set	**Avp**	**Var**	**Std**
A: validation set	1.0699	0.4165	0.5589
B: entire data set	9.0726	75.9549	7.5476

4.2 Training on data set B

The model that was trained on data set B split the images into 70% training, 20%, validation and 10% testing. Again, the trained model was tested against two data sets, the first being the validation set of B and the second being the entire data set of A. The validation set of B consists of 316 images.

In Table 3, the TP, FP, FN and mAP@0.5 of the trained model on data set B can be found. Against validation set B, the model performs strongly with 1118 TP out of the 1264 annotated boxes, 69 FP, 146 FN and a mAP@0.5 of 0.9380. However, in contrast to the model that was trained on A, when looking at the performance of this model against the data set of A, it performs worse. Approximately half of the annotated boxes are not identified. Additionally to this, the model finds 1603 FP and has an mAP@0.5 of 0.4797.

Table 3: Trained model on data set B: TP, FP, FN and mAP@0.5

Data set	TP	FP	FN	mAP@0.5
B: validation set	1118	69	146	0.9380
A: entire data set	1867	1603	2145	0.4797

The pixel distance in Table 4, reflects similar results as in Table 3. Against B set, it performs quite well, however, it has a variance of 6.7525. While, against set A, the average pixel distance is affected by the large variance of 217.5345 pixels.

Table 4: Pixel distance of model trained on set B

Data set	Avp	Var	Std
B: validation set	3.0180	6.7525	2.2504
A: entire data set	16.3147	217.5345	12.7731

4.3 Training on data set B and then A

Both previous models performed poorly when detecting the optical spheres in the other data sets. To see if the model could learn to identify spheres in both data sets. The model was first trained on set B then using the resulting weight, was trained again using set A. The results can be found in Figure 5 and 6.

With an mAP@0.5 of almost 1 and a low number of FP, FN detections, it shows the model performs well against set A, the last set that the model trained on. In contrast to the initial model, where the model performed well on set B, the model after training on set A performs worse on set B, with just over 60% TP detections and a 0.4797 mAP@0.5.

Table 5: Trained model on data set B then A: TP, FP, FN and mAP@0.5

Data set	TP	FP	FN	mAP@0.5
A: validation set	807	4	9	0.9948
B: validation set	772	157	492	0.4797

Looking at the average pixel distance, this model trained on set B and then on A is able to detect the validation set from A with a small average pixel distance. On average, the distance is less than 1 pixel and has a standard deviation of only 0.5327. While, on set B, the model performs better than the model trained on set A, where the variance is lower at 46.2933 but the average pixel distance is at 8 pixels. However, this model still performs poorly compared to the model trained by just data set B.

Table 6: Pixel distance of model trained on set B then set A

Data set	Avp	Var	Std
A: validation set	0.8938	0.3784	0.5327
B: validation set	7.9719	46.2933	5.8924

5 Discussion

In total, 3 different models were tested. The models can be referred to as model A, which was trained on the training set of data set A, model B, which was trained on the training set of data set B and model C, which took model B and trained on the training set of data set A.

The models results use the confidence threshold with the optimal precision and recall values. YOLOv5's validation does this automatically by taking the confidence threshold with the highest F1 score, which takes the harmonic mean between the precision and recall [15]. With the model already be optimised on confidence threshold, it suggests clearly that both model A and B only perform well when validated against their respected validation sets, while when put against the other set the models detect only approximately 50 percent of the annotated spheres, next to detecting a large number of false positives. This can also be reflected in the pixel distance results. Due to the spread in the boxes, the average pixel distance is significantly higher.

Looking at model C, it performed the best out of the 3 models. The model was able to detect spheres from data set A with a high mAP@0.5. However, against data set B, which was trained first, the model had a lot of false and missed detections. The limitation to this model was that it trained consecutively on different data sets. Looking at the results, this suggests that the model was learning to detect the spheres in A and adjusting the weights accordingly and caused the spheres in set B to be forgotten. It would be interesting to verify if the model was trained first on A and then B would give similar results where the model adjusts weights so that the model would perform worse on data set A than B. In all 3 models, the model may be over training, so that the model lacks flexibility when trying to detect spheres that are present in images with a change in background noise. To solve this problem, the model needs to be given a larger variety of data images or a combination of the two existing data sets.

Furthermore, a limitation in data set B was that the 3D positions obtained automatically were not accurate enough. This was because the QR codes were too far away for a stable detection and therefore the spheres were manually annotated, which affected the precision when trying to obtain the 3d

positions. Thus, the bounding boxes in data set B have a fixed bounding box for all spheres. The largest sphere was manually annotated and used as the fixed size for all other spheres. This resulted in bounding boxes that covered more than just the sphere, adding more noise for the model to learn. This limitation suggests the larger pixel distance in model B to model A when tested against their respected validation sets.

Contribution towards HoloNav

The HoloNav project consists of many sub-problems that need to be studied before the use of AR can be seen in surgical navigation. This study has focused on the possible detection of optical reflective spheres with the gray scale cameras of the HL2. Given the results above and the discussion, future works on HoloNav can use this as a starting point to learn more about the possibilities of using YOLOv5, an object detection, deep learning approach to detecting the optical reflective spheres.

Responsible research

To reflect on the ethical aspects of this study. All the results in this study were directly gathered from running the model on the given data set. This data set was randomly split in to a distribution of 70% training, 20% validation and 10% testing. The steps to reproduce these results have been mentioned as thoroughly as possible in this study. However, the results may not be exactly the same as the nature of training; the model contains some randomness. Given the steps in this paper and the data set, similar results should have been gathered if this were to be reproduced.

Next to this, the topic of object detection and AI has gained more and more concern surrounding the topic of ethics. With this being said, the original author of YOLO, Joseph Redmon stopped working on YOLO after YOLOv3 because of "the military application and ethical privacy concerns" [17]. This is crucial when thinking about the future of object detection algorithm for surgical navigation. It is important to reflect on the privacy concerns of potential patients.

6 Conclusion and Future Work

In this study, the possibility of using an object detection algorithm YOLOv5, to precisely detect optical reflective spheres using images from the HL2 gray scale cameras was explored. The results gathered show that YOLOv5, a state-of-the-art object detection algorithm, can be used to precisely detect spheres. However, the results show many limitations that need to be explored further. The trained models showed promising results when tested against validation sets. However, are limited when exposed to the other data set. Additionally, the best performing model was one that was trained on both data sets; however, only performed strongly on the last trained data set. Therefore, it would be interesting to test a data set which was constructed as a combination of both data sets and see if the model is able to adjust the weights to be able to accurately detect spheres in both data sets. Next to that, the second data set could also improve by finding a way to automatically generate accurate bounding boxes around the spheres. Furthermore, it would be interesting to test the performance of YOLOv4 vs that of YOLOv5 on this data set and compare which algorithm performs more accurately.

References

[1] Niessen W.J. Wolvius E.B. et al. Benmahdjoub, M. Multimodal markers for technology-independent integration of augmented reality devices and surgical navigation systems. *Virtual Reality*, 2022.

[2] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *CoRR*, abs/2004.10934, 2020.

[3] Van Gool L. Williams C.K.I. et al. Everingham, M. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, pages 303–338, 2010.

[4] Clemens Dirven Arnaud Vincent Fatih Incekara, Marion Smits. Clinical feasibility of a wearable mixed-reality device in neurosurgery. *World Neurosurgery*, 118:e422–e427, 2018.

[5] Isabelle Guyon. A scaling law for the validation-set training-set size ratio. In *AT T Bell Laboratories*, 1997.

[6] Glenn Jocher. Train custom data · ultralytics/yolov5 wiki, May 2020.

[7] Jeong-ah Kim, Ju-Yeong Sung, and Se-ho Park. Comparison of faster-rcnn, yolo, and ssd for real-time vehicle type recognition. In *2020 IEEE International Conference on Consumer Electronics - Asia (ICCE-Asia)*, pages 1–4, 2020.

[8] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: single shot multibox detector. *CoRR*, abs/1512.02325, 2015.

[9] Uli et al. Mezger. Navigation in surgery. *Langenbeck's archives of surgery*, 398,4(9):501–514, 2013.

[10] Joseph Nelson. Yolov5 is here, Sep 2021.

[11] Upesh Nepal and Hossein Eslamiat. Comparing yolov3, yolov4 and yolov5 for autonomous landing spot detection in faulty uavs. *Sensors*, 22, 01 2022.

[12] Ehab Rahman, Yihong Zhang, Sohail Ahmad, Hafiz Ahmad, and Sayed Jobaer. Autonomous vision-based primary distribution systems porcelain insulators inspection using uavs, 12 2020.

[13] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016.

[14] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015.

[15] Yutaka Sasaki. The truth of the f-measure. *Teach Tutor Mater*, 01 2007.

[16] Jacob Solawetz and Joseph Nelson. How to train yolov5 on a custom-dataset, Jun 2020.

[17] Yuan Yuan. Yolo creator joseph redmon stopped cv research due to ethical concerns, Aug 2020.