Multiclass Object Detection for Surgical Instruments using YOLOv3

Ayush Shetty
Department of Biomedical Engineering
Duke University
Durham, NC 27708
ayush.shetty@duke.edu

Abstract

Before any major surgery, it is the nurse's job to prepare all the important surgical instruments and setup the tray. All instruments must be accounted for immediately after the surgery so that no instrument is left behind inside the patient. Cases of emergency or human error post or mid surgery might cost very vital minutes, which could lead to severe complications or even patient death in a worst-case scenario. This project aims to help the nurses effectively and completely arrange the instrument tray before and after the surgery to account for all the instruments used in the surgery using computer vision.

1 Introduction

Currently nurses prepare a surgical tray using a checklist for the instruments or some of the experienced nurses have it memorized. This leaves a lot of room for human error. A study showed that there are 1-5 cases of a near miss sharp (NMS) where a lost sharp (needle, blade, instrument, guidewire, metal fragment) is recovered prior to the patient leaving the operating room . An average of 21-30 min is spent managing each NMS, making a lost sharp event result in up to 70 min of added OR time [1].

To assist in the OR, in this paper I suggest a vertically down looking camera that clicks pictures of the tray and detects all the instruments present using the YOLOv3 model for object detection using bounding boxes. I believe having an account for all the instruments at all times will not only put the nurse's minds at ease but also decrease the need for managing NMS.

2 Dataset description

The data obtained is from "Labelled Surgical tools and Images" which is a dataset available on Kaggle [2]. The dataset was created for a master's thesis "Sorting Surgical Tools from a Cluttered Tray - Object Detection and Occlusion Reasoning" published in September 2018 by Diana Martins Lavado. The dataset was obtained by clicking multiple photos of each instrument individually and with other instruments using various backgrounds, rotations and inclinations as seen from Figure 1 [3].

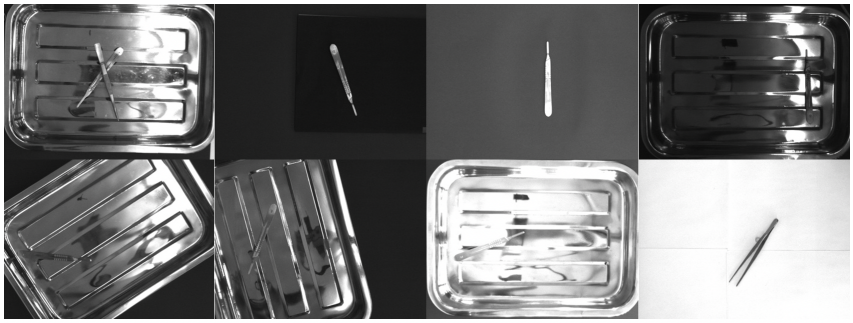


Figure 1: Images in the dataset with various lighting conditions, backgrounds and orientations.

Table 1: Distribution of images based on classes present

Instrument	Individually	With other instruments	All 4 classes present	Total
Scalpel	550	376		1026
Straight Dissection Scissor	460	485	100	1045
Straight Mayo Scissor	550	410		1060
Curved Mayo Scissor	450	479		1029

There are a total of 3009 images in the dataset each of having dimensions 480x640. There are 4 classes of surgical instruments present in the dataset which are: scalpel (denoted class - 0), straight dissection clamp (denoted class - 1), straight mayo scissor (denoted class - 2) and curved mayo scissor (denoted class - 3). The class distribution of the images is fairly even as seen from Table 1. The bounding boxes were labelled using the YOLO mark software. The information for the bounding boxes was saved in the form of a text file with the same name as its respective image file containing the class, normalized center x coordinate, normalized center y coordinate, normalized width and normalized height of the bounding boxes (Figure 2).

```
0 0.479297 0.437500 0.361719 0.058333
1 0.644531 0.528472 0.257812 0.415278
2 0.649609 0.845833 0.383594 0.194444
3 0.287109 0.499306 0.235156 0.548611
```

Figure 2: Sample bounding box text file

3 Related work

The owner of the dataset I am using, Diana Martins Lavado, published a paper in September 2018 where she explored the concern that nurses spend sorting tools after being disinfected, that can either undergo sterilization or be assembled into surgical kits. Thus, that time could be spent focusing more on patients if a robotic system was implemented. She used a YOLOv4 model for object detection purposes achieving a 92% mean precision [3].

A company, RSIP Vision, has been working on surgical instrument (tool) segmentation and classification is a computer vision algorithm that complements workflow analysis. It automatically detects and identifies tools used during the procedure and assess whether they are used by the surgeon correctly [4].

Literature published in 2019 talked about a novel frame-by-frame detection method using a cascading convolutional neural network (CNN) which consists of two different CNNs for real-time multi-tool

detection. An hourglass network and a modified visual geometry group (VGG) network are applied to jointly predict the localization. The results were aimed to help in robot-assisted surgeries [5].

4 Methods

To detect the instruments, I have implemented the 'you only look once' version 3 (YOLOv3) network. YOLO networks are most efficient for detecting objects in real time feed and videos. Every improvement in the version of YOLO is linked with being able to give better performance with higher frame rates. Hence if in future developments, we would want to move the detection system to a real time detector it would be easier which is why I preferred using YOLO.

4.1 Darknet

For this project, I have used the help of darknet to run my YOLO models. Darknet is an open-source neural network framework written in C and CUDA. It is fast, easy to install, and supports CPU and GPU computation. Darknet improves computation and efficiency of object detection algorithms that usually have large amounts of data and need to run for longer epochs. I have used darknet to enable me to utilize the GPU power of my software system to enable training of my model.

4.2 YOLOv3

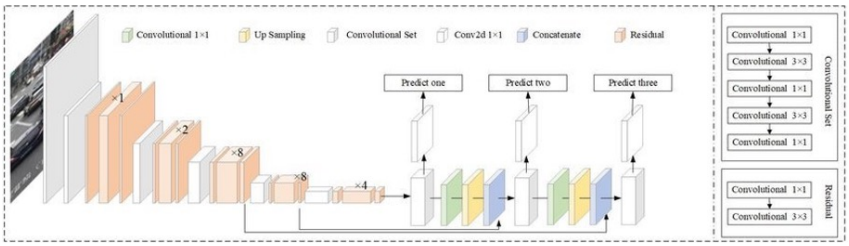


Figure 3: YOLOv3 architecture

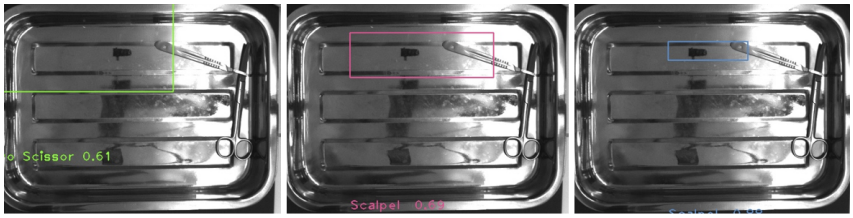
The YOLOv3 algorithm first separates an image into a grid. Each grid cell predicts some number of boundary boxes (sometimes referred to as anchor boxes) around objects that score highly with the predefined classes. Each boundary box has a respective confidence score of how accurate it assumes that prediction should be and detects only one object per bounding box. The boundary boxes are generated by clustering the dimensions of the ground truth boxes from the original dataset to find the most common shapes and sizes. It consists of 3 YOLO layers at the end that are pivotal in making the predictions (Figure 3).

4.3 Model training

The entire dataset was split in a 70:30 ratio for train-test split i.e., 2107 train images and 902 test images. For the model training, I set the batch size to 64 and the number of subdivisions to 16, meaning that each mini-batch had 4 images. The learning rate chosen for the model was 0.001 which was to reduce the computational load on the model for training while trying to maintain a certain level of accuracy. The images were resized to 416x416 when put through training since image sizes need to be a multiple of 32 to pass through the YOLO network.

I also noticed that the image dataset did not contain any blurred images that accounted for camera focusing blurs. Hence, I created a blurred image dataset by using gaussian blur on the images. I trained the weights on these images for 600 iterations to see how they would differ in prediction capabilities.

5 Results



(a) (b) (c)

Figure 4: a) Detection using weights trained for 100 iterations, (b) Detection using weights trained for 600 iterations, (c) Detection using weights trained for 1000 iterations

As seen from Figure 4 (c) the weights trained for 1000 iterations performed the best by drawing a smaller and tighter bounding box close to the instrument itself. The weights trained for 600 iterations (Figure 4 (b)) makes a broader box and also gave lower confidence, while the weights trained for 100 iterations (Figure 4 (a)) gave a very large, inaccurate box with an incorrect detection of the class of the instrument.

The weights trained on the blurred dataset were also tested against the test dataset of the blurred images. The weights performed very poorly in detecting instruments where it did not create any bounding box on the image on the account of making predictions.

6 Discussion

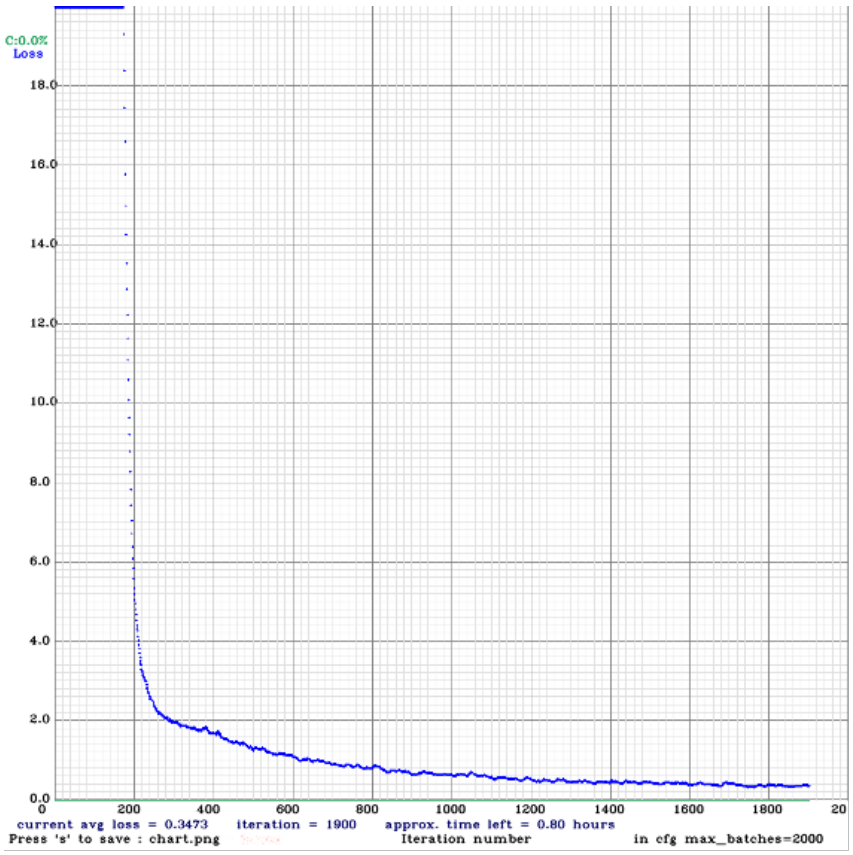


Figure 5: Training loss over multiple iterations

Figure 5 depicts how the training loss reduces over increasing number of iterations. This is what led me to train multiple weight for different iterations to test what difference would it show in the output. Generally, a YOLO network must be iterated for number of classes*2000 iterations which would have been 8000 iterations for my image dataset. Due to lack of computational resources, I could not train my weights for 8000 iterations. In the future with more resources I would like to correctly train the weights to detect the instruments accurately. Since I would also like to detect images post surgery, it would be great if images of instruments with blood could be included in the dataset. Lastly, increasing the number of classes of instruments in the dataset would be essential in detecting more classes of instruments used in a surgery.

7 Conclusion

Images of 4 classes of surgical instruments taken under various lighting conditions, with different backgrounds and orientations were considered for a YOLOv3 object detection. A gaussian blurred image set was used to simulate camera focusing. Altogether, the model trained for 1000 iterations detected the instruments the best in terms of precision of the bounding box and confidence of prediction, while the weights trained on the blurred image set performed the worst. Higher number of iterations would have improved the detection capabilities of the model with availability of more computational resources. Though the model, for the number of iterations used performs poorly, with fine tuning and more training shows promise for application in the operation room.

References

[1] Weprin, S.A., Meyer, D., Li, R. et al. Incidence and OR team awareness of "near-miss" and retained surgical sharps: a national survey on United States operating rooms. Patient Saf Surg 15, 14 (2021). https://doi.org/10.1186/s13037-021-00287-5

[2] Labelled Surgical tools and images: https://www.kaggle.com/datasets/dilavado/labeled-surgical-tools

[3] Lavado, D. M. (2018). Sorting Surgical Tools from a Clustered Tray-Object Detection and Occlusion Reasoning (Doctoral dissertation, Universidade de Coimbra).

[4] Surgical instrument segmentation: https://www.rsipvision.com/surgical-tool-segmentation

[5] Zhao, Z., Cai, T., Chang, F., Cheng, X. (2019). Real-time surgical instrument detection in robot-assisted surgery using a convolutional neural network cascade. Healthcare technology letters, 6(6), 275–279. https://doi.org/10.1049/htl.2019.0064