THE IMPROVEMENT OF OBJECT DETECTION AND LOCALIZATION FOR AUTONOMOUS CAMERA MOVEMENT

A **Capstone Research Paper Submitted to the**

Faculty of the School of Engineering and Applied Science
University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree
Bachelor of Science, School of Engineering

By
Harshneet Bhatia
Spring 2020

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments.

Signature Harshneet Bhatia Date 11/20/2020
Harshneet Bhatia

Approved _____ Date 11/25/2020
Homa Alemzadeh, Department of Electrical and Computer Engineering
(Computer Science by Courtesy)

The Improvement of Object Detection and Localization for Autonomous Camera Movement

Harshneet Bhatia
Advisor: Homa Alemzadeh

***Abstract*—With the rapidly growing presence of robots in the surgical field, the limits of computer vision and deep learning are continuously being tested. This paper focuses on the object detection and classification work previously done for the RAVEN-II and the improvements made to it through augmentation and more generalized labeling. The accuracy and precision metrics have been defined for clarity. The previous overall accuracy for the detection and localization of graspers and multiple blocks was 61.21%, but our most recent model achieved a 99.15% accuracy with the validation dataset.**

I. Introduction

Surgical robots have been in use in the field of medicine since the late 1980s while the origin behind the idea can be dated back as far as 1967 [1]. Robot-assisted minimally invasive surgery (MIS) procedures are becoming more and more common and have enabled procedures with increased precision and dexterity, but some patients and health care professionals remain skeptical of the technology and its reliability [2]. The current MIS setup is in level 0 of autonomy [3]), meaning they are still open loop and require surgeons to work with a teleoperation console providing only limited visual feedback. By increasing flexibility and precision, surgical robots have enabled new types of surgical procedures and have reduced complication rates and procedure times. However, with the proliferation of data collection and data-driven solutions for many traditional problems, there are plenty of improvements that can be made in the field of surgery, such as automating surgical subtasks and reducing the likelihood of accidents during surgery.

The focus of this capstone study was to continue working on the localization and classification of surgical tools and objects of interest in the drylab workspace for the RAVEN
II. The ultimate goal of the project is to achieve state-of- the-art accuracy for surgical workspace perception which can enable autonomous robot manipulation by being an input to motion planning algorithms or improve context-awareness of the robot by providing information about the objects of interest along with their positions. In addition, the study aims to achieve platform independence due to the data modality being vision-based and being orthogonal to the robot's kinematics. We used the same Mask Region-based Convolutional Neural Network (MRCNN) model as before, with ResNet 101 [4] as the backbone. The integration pipeline, steps taken to improve the accuracy results, and how we defined accuracy will be discussed in further detail below.

Contributions. The main contributions of this study are creating the Perception module responsible for the localization and classification of objects for the autonomous camera system and the lasting insights for the usage and training of this MRCNN model. The work can be broken down into the integration of the localization and classification component into an autonomous camera system III-B, verification and improvement of the accuracy of the Perception module III-A, and image augmentation III-C1.

II. Background

Our previous work attempted to address the problem of automated detection and localization of objects in the surgical workspace. There are two separate computer vision tasks here, **Image Classification** where we predict the class of the image and **Object Localization** where we locate the presence of an object in an image and provide their bounding box. Recent success of Convolutional Neural Networks (CNNs) in the task of pattern recognition, and the prominence of the Mask Regional Convolutional Neural Networks (MRCNN) for object detection and localization makes it an ideal fit for our problem. CNNs have been proven to learn rich feature representation, given enough training data, and have achieved state-of-the art results over many pattern recognition tasks. However, as with any Deep Neural Network, we need a large amount (~1M) of strongly labeled training data, which we did not have. Instead we went with the transfer learning approach where we use a pre-trained network that has been trained on a large dataset to extract the higher level features from our images. We fine-tune only the fully connected layers, and replace the final classification layer for our task.

III. Methodology

The methodology can be broken down into three main sections: Object Detection and Localization, Integrating Object Detection into an Autonomous Camera System, and Expanding the Pick and Place Dataset.

A. Object Detection and Localization

The object localization and classification component was previously a stand-alone project. The four objects of interest were the "Right Grasper", "Left Grasper", "Green Block", and "Red Block". The order in which identification occurs is as follows: bounding boxes are generated for each object of interest, then they are classified and a mask is created for each. These objects were identified using an MRCNN model

that has a Residual Networks (ResNet 101) backbone. ResNet acts as a feature extractor, picking up on the key components of an input image, such as the shapes and edges of the objects of interest. ResNet is then followed by the Region Proposal Network (RPN), which examines each region of interest or ROI, before feeding the extracted features into the fine-tuned classification layers. The fine-tuning was done by adjusting hyperparameters, such as the batch size, learning rate, number of epochs, and steps per epoch. The classification layers were pre-trained using the COCO dataset. Through transfer learning, we were able to use the knowledge acquired from previously trained models and apply it to our surgical tasks at hand. COCO was preferred over the ImageNet dataset, due to there being more similarities between the objects that the COCO dataset consists of and the objects of interest in our case. The classification layers are responsible for generating the bounding boxes and masks for each class. The bounding boxes indicate where an object has a high probability of being found in the image, while the masks reveal where the object was actually found. The order of execution for these steps can be seen in Figure 1.

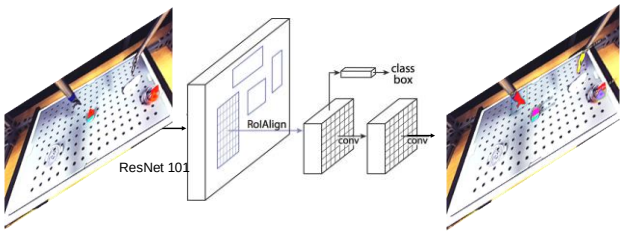


Fig. 1: End-to-end pipeline for detecting and localizing objects in surgical workspace

Data was collected through drylab experiments of the Fundamentals of Laparoscopy (FLS) "Pick and Place" training task using a ZED mini 3D camera. Each image was manually annotated using the VGG Image Annotator (VIA) [5]. The region tool was used to create an outline for each object of interest by using polygon points. Then, the label associated with each object was added to the annotations. The annotated images were used to build upon the classification layers of the NUS Control Mechatronics Lab's implementation [6]. The optimal hyper-parameter choices were determined through adjustments to the training set and fine-tuning of the parameters. We learned that a learning rate of 0.01 and the highest batch size of images supported by the GPU would lead to the most favorable outcome. The model in its final state was able to give the correct bounding boxes for each object, but it consistently gave incorrect maskings for the left grasper and incorrect class labels for multiple objects of interest. In the end, we were able to conclude that further work was required and that image augmentation could be one of the possible enhancements.

B. Integrating Object Detection into an Autonomous Camera System

Last semester we moved towards the integration of the object detection and localization into a larger scaled pipeline with the end goal of being able to adjust the ZED mini based on the location of the found surgical objects of interest. The ZED mini is useful for the purposes of this study, because its dual lenses provide a view that mimics the way a human would view things with its own two eyes. The stereoscopic camera outputs a left and right image for each provided sample. Previously, we dealt with images that were already taken, but we transitioned to the usage of the ZED Mini for real-time detection and localization. The MRCNN model plays a key role within the pipeline. The model calls a detect function which returns a list of dictionaries for each image. The lists contain each of the following: ROI bounding boxes, class ids, scores (representing the probability of the object being found there), and instance masks. For the next stage of the pipeline we are more concerned with the class ids and their bounding boxes, so those are returned as a list of lists. Based on the bounding boxes returned, the location of the objects is then estimated in 3D. These estimations are used to calculate where the centroid would lie. From here, the control module determines how to maximize the area. The camera will zoom in or out, tilt up or down, and pan left or right based on the average distance between the calculated centroid and the location of each detected object. A detailed view of the flow of the pipeline and how it fits in with the Control portion of the autonomous camera movement pipeline can be seen in Figure 2.

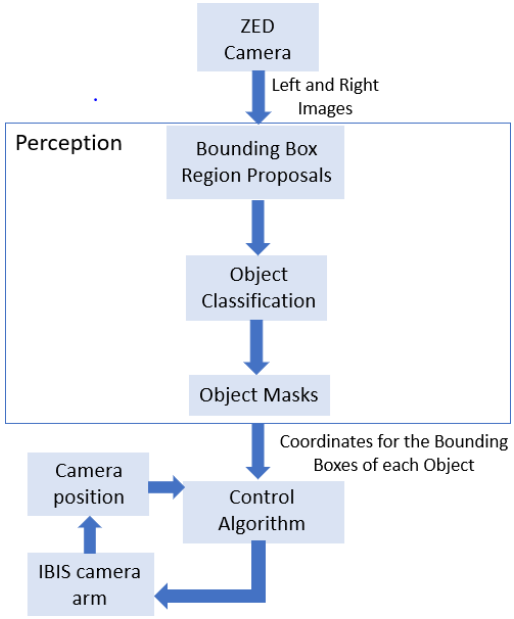


Fig. 2: Autonomous Camera Pipeline

The Control algorithm in the autonomous camera system pipeline does not rely on the classes of objects identified and localized and only needs to have the information about the bounding boxes of the objects of interest. Consequently, we further analyzed the accuracy of the model solely for

identifying the bounding boxes of the objects of interest (blocks and graspers).

C. Expanding the Pick and Place Dataset

Our prior work involved training on a limited dataset, while tuning model hyper-parameters. One of the obvious conclusions was expanding the dataset, as our classification loss was very high.

Our initial dataset consisted of 400 images in the training set and 111 images in the validation set. More labeling was done to bring us to a total of 1686 training images (before augmentation), 600 images for the validation set, and another 600 for the testing set. All 2,886 of these images were taken using the camera. Even after the addition of several hundred images and annotations, more were needed. Since many of the images were very similar to each other, we predicted that the accuracy would improve through expansion of the datasets and diversification of the scenes. Although we were showing each little movement, some objects remained in the same position from one image to the next. The detection of such objects can be difficult, since the model has never seen it in another orientation and position.

1) Image Augmentation: We performed image augmentation to diversify object orientations in the images of the training set and improve the model's ability to detect objects of interest. This is one method of artificially expanding a dataset. Some of the techniques that can be used for image augmentation include scaling, translation, rotation, flipping, adding noise, and changing lighting conditions. The techniques specifically used for augmenting our dataset were flipping the images left/right 50% of the time and generating images with random blending between the original images and their canny edges. These augmentation techniques were only performed on the training dataset. After augmenting, our training dataset reached a total of 60,000 images. This is because the model goes through each image at least once per epoch and we trained for 30 epochs with 500 steps per epoch. The validation and testing dataset totals were kept constant at 600 images each. Canny edge augmentation in itself does not alter the coordinates of the ground truth, but flipping left/right does.

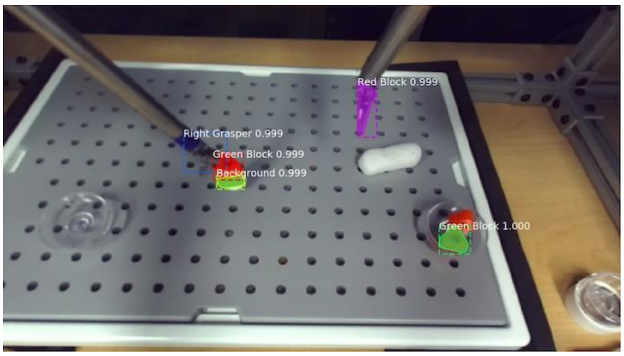
This was an area that raised some concerns. In order to verify that the underperformance of the model was not due to the augmentation and/or the augmented labels being generated incorrectly, we first saved the augmented images and then we saved some of the augmented images along with their corresponding bounding boxes, masks, and labels displayed. Through inspection of the latter, we were able to conclude that the labels were, in fact, the issue. However, the original images and their ground truth were correct. From here, we decided to train with a smaller subset of the original classes. The labels for each object in the JSONs were either "Grasper" or "Block" (eliminating the color and positioning of the blocks and graspers, respectively).

IV. Results

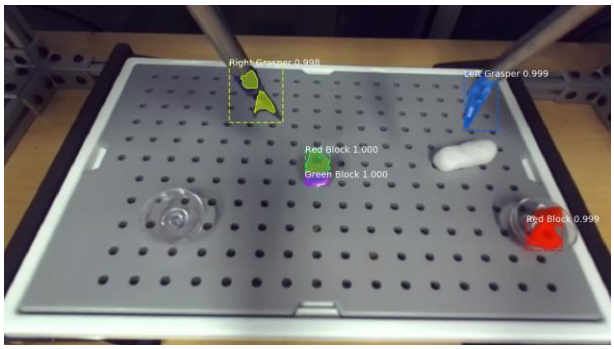
Previous work did not include a measure of accuracy to report on, just losses. Consequently, last semester we worked on coming up with metrics for each MRCNN model training, because the losses decreasing does not necessarily indicate that our model is performing well. The model could be learning incorrect class labels and still showing lower losses. The accuracy and precision were calculated for the new MRCNN model. These metrics were derived by determining whether or not the masks and labels of the model's prediction lined up with the ground truth of each image in its provided JSON. The precision was a measure of how often all of the labels and masks were correct. This was calculated for each image in the testing dataset and then averaged. Table I displays the types of techniques used, the total number of images generated using that technique, and the resulting accuracies and precision rates. Expansion of the RAVEN II's dataset through annotations and image augmentation gave us better results in previous semesters. However, we were able to see better maskings and classification for overlapping objects when we reduced the annotations set to only three labels: "Background," "Grasper," and" Block". Figure 3 shows the resulting localization and classification at different stages. Figure 3a displays our initial results. At this point in time, we had just 400 images in our training set and 111 images in the validation set. Figures 3b and 3c show the results achieved after increasing the training dataset in size and performing image augmentation. The training set had reached 1686 images before augmentation and 60,000 after augmentation. Earlier we were under the impression that the dataset would just double in size after augmentation, but further investigation proved that this was an underestimation. The validation set was at 600 images for the results produced in Figures 3b-3d. Figures 3b and 3c were a result of training with the more specific class names and Figure 3d was a result of the generalized class names. Figure 3b was obtained after training for 20 epochs, while Figure 3c was obtained after training for 30 epochs.

From Figure 3a we can see that all objects are being localized with the exception of the Left Grasper. However, there is only a bounding box in the region of the Left Grasper and all objects are misclassified. After fixing some of the indexing issues found in the code, the classification became significantly better as apparent in Figure 3b. After training for more epochs, the mask for the Left Grasper appears to cover more of the appropriate area in Figure 3c, but the labels for right and left graspers were consistently flipped. Hence, for the next approach we decided to use the broader classes of graspers and blocks and with that we saw improvement in the model's performance as depicted in Figure 3d. It is worth noting that in this training, there is better detection in occluded object scenarios.

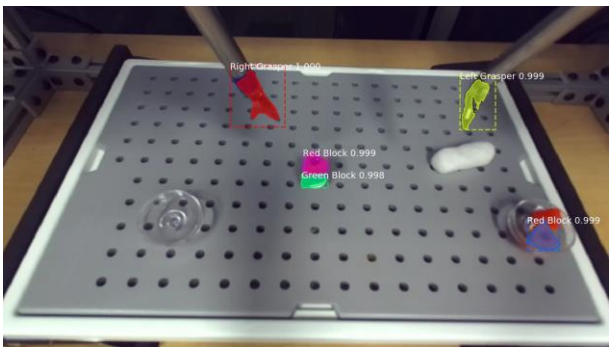
The augmentation techniques were broken down to one per training before combining both of them to see if the type of augmentation technique(s) used would impact the accuracy and/or precision. This made little to no impact in that regard. After replicating and verifying the results that were reported in our International Symposium on Medical Robotics (ISMR 2020) paper [7], we decided to use both the flipping left/right 50% of the time and canny edges augmentation while using



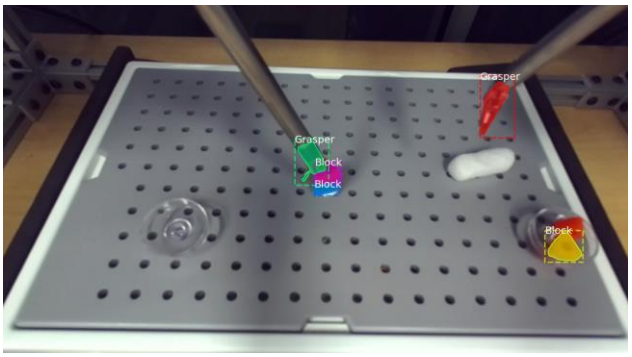
(a)



(b)



(c)



(d)

Fig. 3: Resulting Localization and Classification

TABLE I: Accuracy and Precision for Classification based on Repository Metrics

Augmentation Technique	Total Number of Images	Accuracy	Precision
None	1686	45.31%	44.05%
Flip Left/Right 50% of the time	60044	45.37%	43.84%
Canny Edges	60242	45.35%	44.07%
Flip L/R + Canny Edges[1]	30105	45.32%	44.35%
Both + Generalized Labels	60261	45.31%	43.93%

[1]Trained with a lower batch size, due to memory errors occurring in Google Colab (images per gpu was adjusted from 4 to 2)

a subset of the class labels from before. Once again, the numbers were consistent when computing the accuracy and precision for the testing dataset. However, when performing the computations on the validation dataset, the numbers were much higher. The accuracy and precision for this were **99.15%** and **88.56%**, respectively. This suggests that the model is over-fitting on the training dataset and that the lower performance is a direct cause of it. Future work could focus on experimenting with other augmentation techniques, computing the difference in diversity amongst sets, and reducing the number of parameters to learn so that overfitting is prevented. Other options may include dropout, the early stopping technique on the validation dataset, and perhaps feeding in hints to the model if we would prefer for it to be able to use the more specific labels.

We also found other ways to quantify our results and report on them in our ISMR paper. This was done through the analysis of 27 left and right images taken by the ZED Mini while executing the "Pick and Place" surgical task. The objects of interest were localized and classified with our MRCNN model and these same 54 images were manually annotated so that there would be a ground truth to compare our model's results to. Since the left and right graspers were flipped in almost every image, we decided to see if the model was correctly picking up on graspers in general. The same was done for the color of the blocks. An overall accuracy, graspers accuracy, and blocks accuracy were calculated by taking the ratio of the correctly classified objects and the number of objects present in the ground truth (see Table II). The highest accuracy of the three was for the graspers accuracy at 80.77%. The classification accuracy for the blocks was 52.87% and the overall accuracy for all objects was 61.21%. This did not account for how well the masks matched the annotations of the JSON, since we were only checking if the center of the predicted mask was within the ground truth's bounding box.

TABLE II: Object Classification Accuracies

Class	Correctly Classified	Total	Accuracy (%)
Graspers	84	104	80.77
Blocks	129	244	52.87
All objects	213	348	61.21

Based on the predicted bounding boxes, masks, and labels seen, we anticipate that the accuracies using this same metric would be higher than what we saw in the past. Due to limited resources and unforeseeable circumstances, the computations for this could not be completed.

In terms of performance metrics for the model, there may be a better way to measure how well or poorly the model is

performing. The classification accuracies obtained by taking a look at the ratio of objects classified correctly and the objects labeled in the ground truth were a good approach, but the set of images this was based on was limited in size. This method requires additional annotating and can be quite time-consuming with a larger dataset. Mean Average Precision (mAP) is another metric used in accuracy calculations. It is computed by taking the average of the area under the recall-precision (RP) curve [8]. This might be a better indicator of our model's performance.

V. Discussion and Analysis

Our MRCNN model's performance has improved considerably despite the COVID-19 situation and the impact it had on our research progress. Some of the challenges we faced were Google Colab specific (runtime disconnects, memory errors, etc.), while others pertained to the limited access we had to the lab machines and the RAVEN. In light of all this, the localization and classification of our objects of interest is closer to being correct. This capstone has provided me with several discussion points, which I will address in the remainder of this section.

That being said, there is room for further improvement and the metrics need to support what we have seen through the progression of our resulting images. The graspers appear to be blending in with the background and although apparent to the naked eye, the two can be easily confused with a camera. Better contrast and lighting could potentially resolve this. The ZED mini was positioned in a way such that the area surrounding the surgical workspace was still in the field of view. Another contributor to lower accuracies could be the edges of the objects in the images being poorly defined. A more crisp image with sharper detail will lead to better classification and localization. This can be attributed to the focal length of the stereoscopic camera.

Our most recent results suggest that overfitting might be occurring. We based this off of our 99% accuracy on the validation set and the lower numbers we saw for the testing set of our last training. Typically, when a model performs well in its training and validation, but not so well in testing, this is because the MRCNN model has been overfitted and can not adapt as well to scenarios not seen before. Lastly, another possible cause could be the lack of substantial difference between the original images and the augmented images. Some augmentations may result in more drastic change than the ones we used.

VI. Conclusion and Future Work

This capstone study focused on the improvement of the localization and classification of certain objects of interest for the Perception module of a pipeline for an autonomous camera system. We made improvements to the performance of our MRCNN model through image augmentation and more generalized labeling.

Our final MRCNN model showed better classification and localization of objects through the prediction output images. We expect the ISMR accuracy metrics to reflect this as well (the overall accuracy using this metric earlier was 61.21%). Using this trained model, we saw a validation set accuracy of 99.15% and a precision of 88.56%. The improvements in the model's performance are promising, but future work would yield better results. Our image augmentation techniques included flipping left/right and blending canny edges, but there are other techniques available. Some of them might be able to introduce more diversity in the augmented dataset compared to the original dataset. One such technique might be copy-and-paste augmentation. A way of measuring the difference between the original and augmented images would also be useful here. Adjusting the hyperparameters, since the training conditions no longer remain the same after augmentation could be of interest as well. This could resolve the overfitting, if that is the case. Future work may include incorporating either or both of these potential strategies and observing the effect that they have on the overall performance of the model.

REFERENCES

[1] E. I. George, C. T. C. Brand *et al.*, "Origins of robotic surgery: from skepticism to standard of care," *JSLS: Journal of the Society of Laparoendoscopic Surgeons*, vol. 22, no. 4, 2018.

[2] U. Leung and Y. Fong, "Robotic liver surgery," *Hepatobiliary surgery and nutrition*, vol. 3, no. 5, p. 288, 2014.

[3] G.-Z. Yang, J. Cambias, K. Cleary, E. Daimler, J. Drake,
P. E. Dupont, N. Hata, P. Kazanzides, S. Martel, R. V. Patel *et al.*, "Medical robotics—regulatory, ethical, and legal considerations for increasing levels of autonomy," *Sci. Robot*, vol. 2, no. 4, p. 8638, 2017.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learn- ing for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[5] A. Dutta and A. Zisserman, "The vgg image annotator (via)," *arXiv preprint arXiv:1904.10699*, 2019.

[6] S. Ye. Surgery robot detection segmentation. [Online]. Available: https://github.com/SUYEgit/Surgery-Robot-Detection-Segmentation

[7] K. Hutchinson, M. S. Yasar, M.S., H. Bhatia, and H. Alemzadeh, "A Reactive Autonomous Camera System for the RAVEN II Surgical Robot," To appear in the International Symposium on Medical Robotics (ISMR), 2020. arXiv preprint arXiv:2010.04785.

[8] K. Oksuz, B. Can Cam, E. Akbas, and S. Kalkan, "Localization recall precision (lrp): A newperformance metric for object detection," *arXiv e-prints*, p. arXiv:1807.01696, Jul 2018.