Detection of Surgical Instruments Using Faster RCNN & YOLO: An Enhanced Approach

Shubham Laxmikant Deshmukh
shubhamd23@vt.edu
Virginia Polytechnic Institute and State University
Falls Church, Virginia, USA

Pradyumna Kombethota Ramgopal
pradyumna@vt.edu
Virginia Polytechnic Institute and State University
Falls Church, Virginia, USA

Abstract

The integration of computer vision into the medical field has significantly advanced surgical assistance and automated monitoring. Artificial intelligence (AI) in medical surgeries aims to automate workflows and provide real-time feedback to surgeons, thereby enhancing efficiency and potentially saving patients' lives. Accurate detection of surgical instruments during procedures is critical for improving operational precision and ensuring patient safety.

This paper focuses on the detection of surgical instruments in neurosurgical training exercises using state-of-the-art object detection deep learning architectures, Faster R-CNN and YOLOv8. While prior methods such as RetinaNet and AutoML have been explored for similar tasks, our study provides a detailed comparative analysis between Faster R-CNN and YOLOv8. Although Faster R-CNN is known for its high accuracy, its performance in real-time detection is limited. Conversely, YOLOv8 demonstrates strong real-time detection capabilities alongside competitive accuracy.

Our experimental results reveal that YOLOv8 outperforms Faster R-CNN, achieving a mean Average Precision (mAP) of 76.9% compared to 62% for Faster R-CNN. These findings highlight the potential of YOLOv8 as a robust and efficient tool for surgical instrument detection, making it well-suited for real-time applications in surgical environments.

Keywords: Deep learning, Computer vision, Surgical instrument detection, Faster R-CNN, YOLOv8, Object detection, Real-time systems, mAP

ACM Reference Format:
Shubham Laxmikant Deshmukh and Pradyumna Kombethota Ramgopal. 2018. Detection of Surgical Instruments Using Faster RCNN & YOLO: An Enhanced Approach. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX).* ACM, New York, NY, USA, 10 pages. https://doi.org/XXXXXXX.XXXXXXX

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Deep Learning has seen significant advancements over the past decade, especially with the advent of architectures such as Convolutional Neural Networks (CNNs) for image recognition and processing tasks. These advancements have enabled breakthroughs in computer vision, leading to widespread adoption in various domains, including healthcare. One particularly important application of deep learning in healthcare is the detection and classification of surgical instruments during operations. Accurate and efficient detection is critical for ensuring patient safety, enhancing operational efficiency, and minimizing the risk of complications during surgeries.

The motivation for detecting surgical instruments stems from the need to automate parts of surgical workflows, enabling real-time feedback to surgeons, improving tool tracking, and assisting in robotic surgeries. While conventional methods, including feature-based and machine learning approaches, have been employed, they often fall short in terms of accuracy and speed. This is particularly the case when detecting multiple instruments simultaneously or when dealing with complex surgical environments. Modern object detection frameworks, like YOLO (You Only Look Once), have shown great promise in overcoming these limitations, offering real-time processing and high detection accuracy.

Our work is primarily focused on detecting neurosurgical instruments using the Simulated Outcomes Following Carotid Artery Laceration (SOCAL) dataset, which provides a wide range of annotated images depicting surgical tools used during neurosurgery [1]. The dataset contains 365 trials from 177 surgeons, with 31,443 annotated frames from 147 video trials. It includes detailed annotations for eight surgical tools, such as suction, grasper, cottonoid, muscle, drill, scalpel, string, and tool under various conditions. Each frame is labeled with bounding boxes for visible tools, offering a challenging dataset for detecting and identifying surgical instruments in complex environments. The dataset provides video frames in JPEG format and corresponding annotations in CSV format, making it ideal for training and evaluating computer vision models in surgical tool detection.

The SOCAL dataset is particularly challenging due to the high degree of variability in instrument positioning, lighting conditions, and occlusions. Furthermore, the dataset includes instances where multiple instruments overlap, or partial views of instruments are obstructed by surgical hands, making detection a complex task. These factors provide an excellent test bed for evaluating the robustness of computer vision models in real-world surgical environments.

In this study, we propose to evaluate and compare the performance of two state-of-the-art object detection architectures, Faster R-CNN and YOLO, for detecting neurosurgical instruments in the SOCAL dataset. Faster R-CNN, a two-stage detector, is known for its high precision, particularly in complex scenarios involving small or overlapping objects. However, its inference speed tends to be slower, which can be a limitation in real-time applications. On the other hand, YOLO (You Only Look Once), a single-stage detector, is designed for real-time object detection with faster processing speeds, but it may compromise accuracy when detecting smaller or partially occluded objects. By comparing these models on the SOCAL dataset, which presents challenges such as variable instrument positioning, occlusion, and lighting conditions, we aim to provide a comprehensive evaluation of the trade-offs between detection speed and accuracy. This comparison will offer insights into the suitability of each model for real-time surgical instrument detection, highlighting the advantages and limitations of both architectures in surgical environments.

2 List of Contributions

Our project build upon prior efforts to detect surgical instruments using advanced object detection techniques, specifically Faster R-CNN and YOLO to neurosurgical data. A significant enhancement in our approach involves the use of transfer learning to improve the performance[10]. As training a model from scratch could be computationally expensive and inefficient, we initialize our model with weights pre-trained on large scale object detection datasets such as COCO. This transfer learning approach allows the model to leverage the general object recognition knowledge gained from the datasets and fine-tune it for the specific task of neurosurgical instrument detection.

Our primary contribution lies in evaluating and comparing the performance of two well-known object detection architectures—Faster R-CNN and YOLO—on this dataset. We focus on identifying the model that achieves higher mean Average Precision (mAP) scores, while also introducing new approaches that have not been previously applied to this dataset. Additionally, we benchmark our results against prior methods, demonstrating significant improvements in both precision and recall, particularly for detecting complex surgical instruments. By leveraging transfer learning, we enhance the model's ability to efficiently detect instruments in the challenging environment of surgical procedures. This approach not only accelerates the detection process but also enhances the model's scalability for real-time surgical applications.

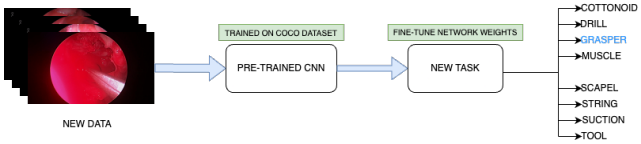

Figure 1. Transfer Learning Process for YOLO Model

3 Related Work

In previous studies, deep learning models such as RetinaNet and AutoML[9] have been explored for detecting surgical tools. RetinaNet[7], while achieving good precision, often struggles in real-time applications due to its relatively slower inference speed, especially on large, high-resolution datasets like SOCAL. Similarly, AutoML[8], which automates the process of model selection and optimization, has shown promise but lacks the fine-tuning capabilities necessary for datasets with complex instrument occlusions and variations [2].

Some studies have also implemented transfer learning on well known object detection algorithm called YOLO (You look only once)[6]. YOLO model is really famous for fast real time object detection and with less computational power. It is also easy to code for transfer learning on custom dataset. One research [3] have implemented YOLOv4 on a similar dataset we have with a similar set of classes and they achieved a good mAP of 87.5.

Further improvements to the YOLO architecture, such as YOLOv7, have also been applied to the task of surgical instrument detection, achieving even higher accuracy. For instance, YOLOv7 has been shown to achieve a mAP of 95.8% in real-time instrument detection applications, making it one of the most promising models for this task [4]. Another study compared multiple models using IBM's Visual Analytics tool, reporting an accuracy of 91% for surgical instrument detection, further emphasizing the effectiveness of deep learning models in this domain [5].

4 Data Description

As part of this study, we collected a dataset consisting of a folder of images totaling 3.2 GB in size. The corresponding annotations for each image are stored in the SOCAL.csv file, where each row contains the image path, along with the x, y coordinates, width, height parameters, and the class label in the final column. The dataset includes eight distinct surgical tools, with the following distribution: cottonoid (10,005 annotated instances), drill (210), grasper (15,943), muscle (4,560), scalpel (4), string (11,917), suction (22,356), and a general tool class (76). To facilitate model training, we have converted

this dataset into a COCO-friendly format, where each image is associated with a corresponding text file that contains the class label and its x, y coordinates with the width and height of the bounding boxes. This conversion allows us to easily utilize state-of-the-art object detection models like Faster R-CNN and YOLO for further analysis.

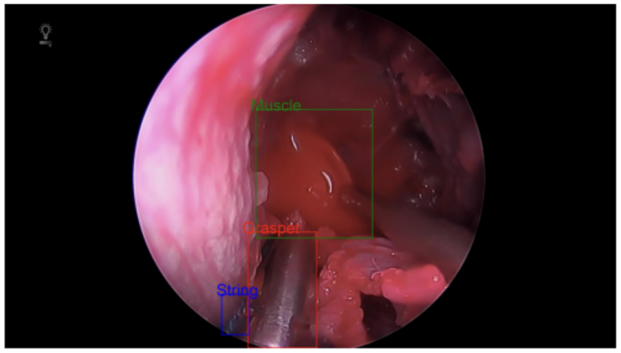

Figure 2. A sample image from the dataset, showing the bounding boxes and labels

For training purposes, we used a subset of the dataset collected from the source [11]. The training dataset comprises 960 images along with their corresponding labels, while the validation dataset contains 220 images and their labels. The dataset consists of images in the '.jpeg' format, with their corresponding labels stored in '.txt' files. The label format is illustrated in Figure 4.

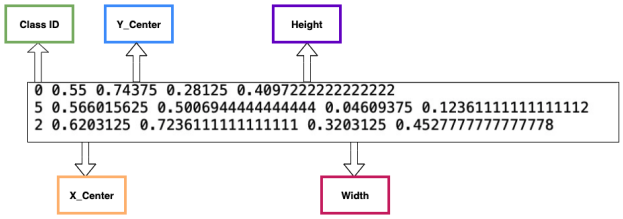

Figure 3. Annotated data in the YOLO format

Each label file follows the format: the first number represents the class label, followed by the x-coordinate and y-coordinate of the bounding box, and then the width and height of the bounding box. There are 8 distinct labels, which are mapped in the following order: ['cottonoid', 'drill', 'grasper', 'muscle', 'scalpel', 'string', 'suction', 'tool'].

An image may contain multiple labels, indicating the presence of multiple bounding boxes within the same image. Consequently, each '.txt' file can include multiple entries, corresponding to the different bounding boxes and their respective labels within a single image.

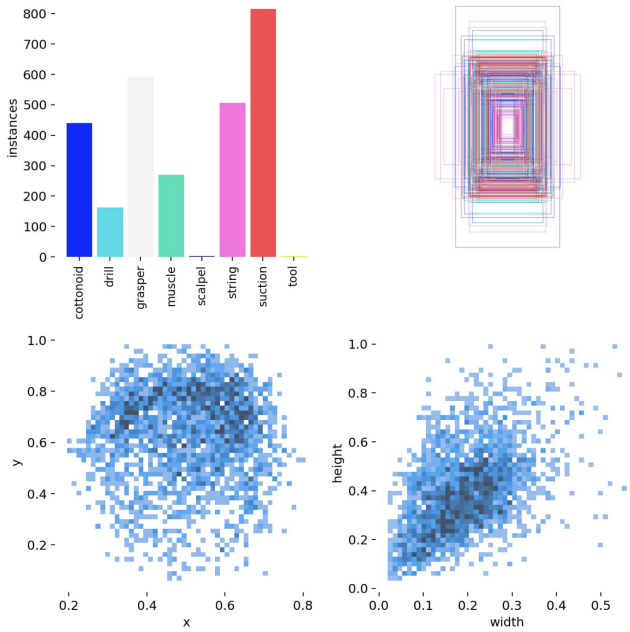

Figure 4. Dataset insights: category counts, bounding boxes, spatial distribution, and size correlations.

5 Model Description

In this study, we leverage transfer learning techniques to enhance the performance of object detection on our dataset. Transfer learning is a machine learning approach where a model trained on a large, general-purpose dataset is adapted to a specific, smaller dataset. This technique allows the model to utilize previously learned features, reducing training time and improving accuracy, particularly in scenarios with limited data availability.

We employ two state-of-the-art deep learning architectures: Faster R-CNN and YOLOv8, both fine-tuned to detect surgical instruments across eight distinct classes: cottonoid, drill, grasper, muscle, scalpel, string, suction, and tool.

5.1 Faster R-CNN

Faster R-CNN is a two-stage object detection framework that integrates a Region Proposal Network (RPN) with a Fast R-CNN detector. This architecture combines the strengths of region-based methods and convolutional neural networks (CNNs) to achieve accurate and efficient object detection [13]. The key stages of Faster R-CNN are as follows:

- **Feature Extraction:** A convolutional neural network (CNN) backbone, such as ResNet-50, extracts hierarchical feature maps from the input image.
- **Region Proposal Network (RPN):** The RPN scans feature maps to generate bounding box proposals likely to contain objects using anchor boxes and regression offsets.

- **RoI Align:** Region proposals are aligned to a fixed size using bilinear interpolation, addressing spatial misalignments and preserving feature precision.
- **Classification and Regression:** The aligned regions are classified into object categories, and bounding box coordinates are refined.

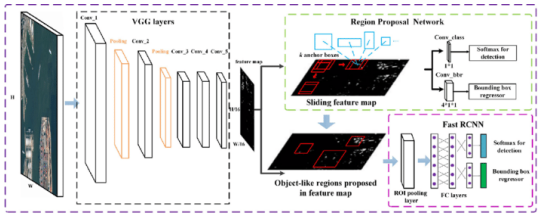

Figure 5. Architecture of Faster R-CNN showing the feature extraction, RPN, RoI Align, and classification stages. Adapted from [15].

ResNet-50 Architecture. ResNet-50 serves as the backbone for Faster R-CNN, leveraging a residual learning framework to mitigate the vanishing gradient problem in deep networks [12]. Its architecture includes:

- **Convolutional Layers:** The network starts with a 7x7 convolutional layer followed by max-pooling, extracting basic features.
- **Residual Blocks:** Skip connections (shortcuts) allow gradients to flow through the network, enabling the training of deeper architectures.
- **Bottleneck Blocks:** Each block consists of three layers (1x1, 3x3, and 1x1 convolutions) to balance computational efficiency with representation capacity.
- **Feature Pyramid Networks (FPN):** Integrated FPN enhances multi-scale feature extraction, facilitating the detection of objects of varying sizes.

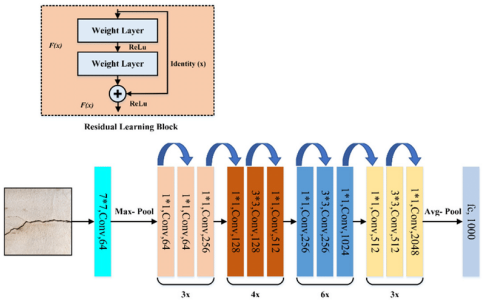

Figure 6. ResNet-50 architecture with its convolutional layers, bottleneck blocks, and residual connections [16].

Implementation Details. The Faster R-CNN implementation is based on the PyTorch `torchvision.models.detection` module, initialized with pre-trained weights on the COCO dataset [6]. To adapt the model to our dataset, the final classification head was replaced to accommodate eight object classes and a background class. The modified architecture is illustrated in Figure 7, where the original head, designed for 80 classes in the COCO dataset, is replaced with a new head specifically tailored for our dataset's 8 object classes. This modification involves reconfiguring the fully connected layers and the softmax output layer.

Key components of the Faster R-CNN architecture include:

- **Backbone:** ResNet-50 with FPN for robust multi-scale feature extraction.
- **RPN:** Generates anchor-based region proposals for refinement.
- **RoI Align:** Ensures consistent feature pooling for accurate region alignment.
- **Modified Classification Head:** A fully connected layer followed by a softmax function, specifically configured for 8 object classes and one background class.

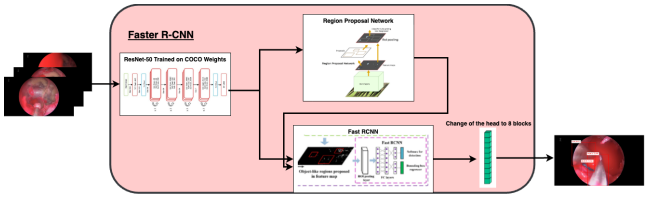

Figure 7. Modification of the Faster R-CNN architecture to adapt the classification head for detecting 8 object classes in our dataset. The original COCO head (80 classes) is replaced with a customized head (8 classes and background).

Training and Evaluation. The model was trained using stochastic gradient descent (SGD) with the following hyperparameters:

- **Learning Rate:** An initial learning rate of 0.005, reduced by a factor of 0.1 every three epochs using a step scheduler.
- **Momentum:** Set to 0.9 to stabilize updates and accelerate convergence.
- **Weight Decay:** Set to 0.0005 to penalize large weights and prevent overfitting.

To further optimize training and prevent overfitting, early stopping was implemented with a patience value of 3. This ensured that training would halt if the validation loss did not improve for three consecutive epochs, saving computational resources and avoiding unnecessary overfitting.

Data Augmentation. To enhance generalization, the Albumentations library was employed for data augmentation. The augmentation pipeline included:

- **Horizontal Flipping:** Random horizontal flips with a probability of 0.5.
- **Brightness/Contrast Adjustment:** Random brightness and contrast changes with a probability of 0.2.
- **Scaling, Rotation, and Translation:** Random scaling, rotation, and shifting within ±5% limits, applied with a probability of 0.5.

These augmentations improve the model's robustness to variations in the input data [14].

5.2 YOLOv8

YOLOv8 (You Only Look Once version 8) is a single-stage object detection model known for its speed and efficiency. It uses a unified architecture with improved feature pyramid networks and decoupled heads for classification and regression, enabling accurate detection at various scales. The process of YOLO can be broken down into several steps:

- **Feature Extraction:** Input image is passed through a Convolutional Neural Network to extract key features from the image.
- **Class and Bounding Box Predictions:** The extracted features are then passed into fully connected layers that produce predictions for object classes as well as the coordinates of the bounding boxes.
- **Intersection over Union (IoU):** The Intersection over Union (IoU) metric quantifies how closely a predicted bounding box aligns with the actual ground truth box.
- **Non-Max Suppression (NMS):** A post-processing step known as non-maximum suppression is applied to remove redundant, overlapping bounding boxes, retaining only the most confident predictions.

One of the key benefit of YOLO is that it processes the entire image in a single pass, making it faster and efficient, when compared to R-CNN and its variants. The YOLOv8 architecture can be broadly divided into three main components:

- **Backbone:** This is the convolutional neural network (CNN) responsible for extracting features from the input image.
- **Neck:** The neck, also known as the feature extractor, merges feature maps from different stages of the backbone to capture information at various scales.
- **Head:** The head is responsible for making predictions.

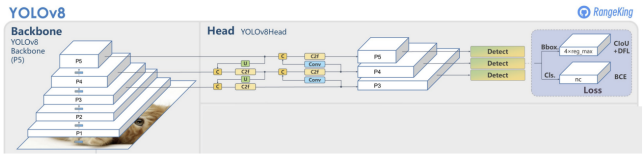

Figure 8. YOLOv8 Architecture [17].

Implementation Details. The YOLOv8 implementation is based on the Ultralytics library. We used YOLOv8x pretrained weights that were trained on the COCO dataset. To adapt the model to our dataset, we replaced the final classification head to accommodate eight object classes. This modification involved reconfiguring the fully connected layers and the softmax output layer. The resulting model was then trained for 150 epochs.

Training and Evaluation. The model was trained using AdamW optimizer with the following hyperparameters:

- **Learning Rate:** The learning rate is set to 0.0008.
- **Momentum:** Set to 0.9 to stabilize updates and accelerate convergence.
- **Weight Decay:** Applied differently to various parameter groups.

6 Evaluation and Experimental Results

To assess the effectiveness of our model, we plan to take the following steps

6.1 Evaluation Metrics

We will evaluate our model using standard performance metrics in object detection, including:

- **Mean Average Precision (mAP)**: It is calculated by finding Average Precision(AP) for each class and then average over a number of classes.

$$\text{mAP} = \frac{1}{N}\sum_{n=1}^{N} \text{AP}_n \tag{1}$$

- **Precision**: It measures how often our model correctly predict a target class.

$$\text{Precision} = \frac{T_p}{T_p + F_p} \tag{2}$$

- **Recall**: It measures whether our model can find all the objects of the target class.

$$\text{Recall} = \frac{T_p}{T_p + F_n} \tag{3}$$

- **F1-Score**: This is calculated as the harmonic mean of the precision and recall scores.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

6.2 Faster RCNN Results

The Faster RCNN model's performance was evaluated using multiple metrics, including mean Average Precision (mAP), precision, recall, and F1-score. These metrics provide a comprehensive evaluation of the model's object detection capabilities:

- **Mean Average Precision (mAP)**: The mAP measures the model's ability to accurately detect objects by computing the area under the precision-recall curve for

each class and averaging across all classes. In our experiments, the Faster RCNN model achieved a peak mAP of **0.6258** at an Intersection over Union (IoU) threshold of 0.5. This indicates effective detection across multiple object classes. Figure 10 illustrates the mAP progression over training epochs, showing steady improvements during training, with both training and validation mAP stabilizing after epoch 6.

- **Precision**: The model achieved an overall precision of **0.32**, which indicates a moderate ability to filter out false positives. This metric is essential in applications where minimizing incorrect detections is critical.
- **Recall**: The model achieved an overall recall of **0.32**, reflecting its capability to detect relevant objects consistently.
- **F1-Score**: The model achieved an F1-score of **0.3150**, demonstrating a balance between precision and recall in its predictions.

To evaluate the model's training process, we analyzed the training and validation loss curves (Figure 9) and mAP progression curves (Figure 10). The loss curves demonstrate steady optimization throughout the training epochs, while the mAP curves show consistent improvements in detection performance, stabilizing after epoch 6.

The confusion matrix provides further insights into the model's classification performance across the eight surgical instrument classes. Figure 11 illustrates the distribution of predictions versus actual labels.

Observations from the confusion matrix:

- Class 7 (*suction*) shows the highest detection accuracy with 69 correct predictions, indicating the model's ability to identify this class effectively.
- Significant misclassifications are observed between Class 3 (*grasper*) and other classes, particularly with 45 predictions misclassified as Class 7.
- Class 5 (*scalpel*) demonstrates poor detection performance, with no correctly identified instances, reflecting the challenges in detecting smaller or underrepresented objects.
- Overlaps and occlusions likely contribute to confusion between certain classes, such as Class 6 (*string*) and Class 7 (*suction*).

6.3 YOLOv8 Results

The YOLOv8 model's performance was evaluated using multiple metrics, including mean Average Precision (mAP), precision, recall, and F1-score.

- **Mean Average Precision (mAP)**: In our experiment, the YOLOv8 model achieved a mAP of **0.769** at an Intersection over Union (IoU) threshold of 0.5.
- **Precision**: The model achieved an overall precision of **0.788**.
- **Recall**: The model achieved an overall recall of **0.722**.

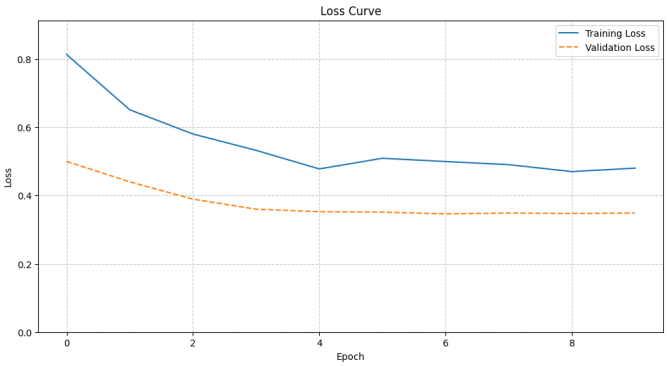

Figure 9. Training and Validation Loss Curves. The model demonstrates steady optimization, with validation loss stabilizing after epoch 6.

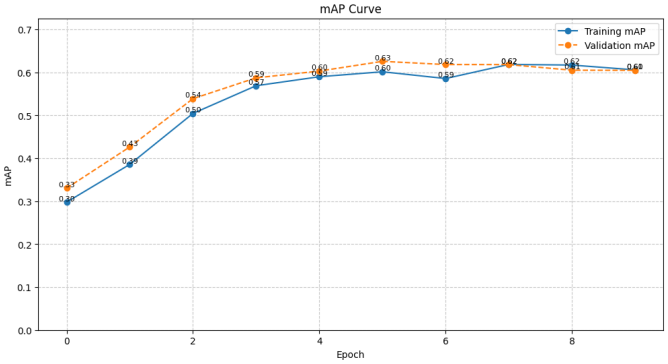

Figure 10. Training and Validation mAP Curves. mAP progression indicates consistent improvements, stabilizing after epoch 6.

- **F1-Score**: The model achieved an F1-score of **0.75**.

6.4 Comparative Results

The comparative analysis presented in the table highlights the performance of several object detection models measured by mean Average Precision (mAP), a standard metric for evaluating object detection systems. This analysis provides valuable insights into the relative strengths and weaknesses of each model, helping to identify the most effective approach for surgical instrument detection.

Our implementations of YOLOv8 and Faster R-CNN achieved mAP scores of 0.769 and 0.625, respectively. These results demonstrate that YOLOv8 outperforms Faster R-CNN, setting a new performance benchmark among the evaluated models. YOLOv8's higher mAP highlights its superior detection capabilities, particularly in scenarios requiring real-time performance without compromising accuracy. On the other hand, while Faster R-CNN shows good precision, its relatively lower mAP reflects its limitations in handling faster

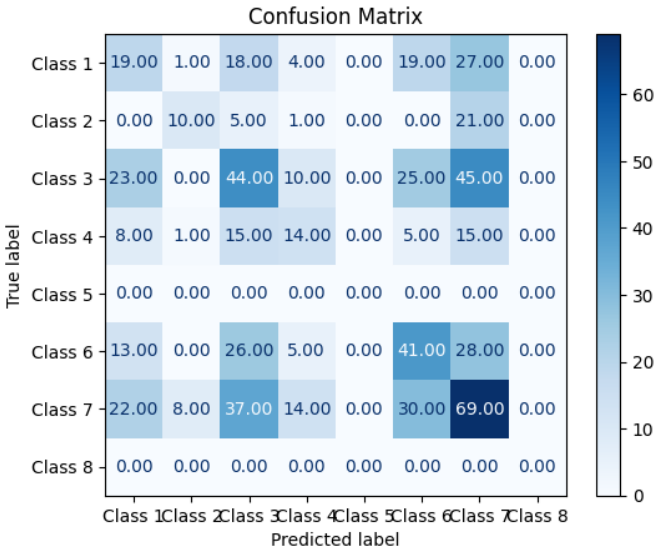

Figure 11. Confusion Matrix (Faster R-CNN). The matrix visualizes the classification performance across the eight classes, highlighting areas of strength and misclassification.

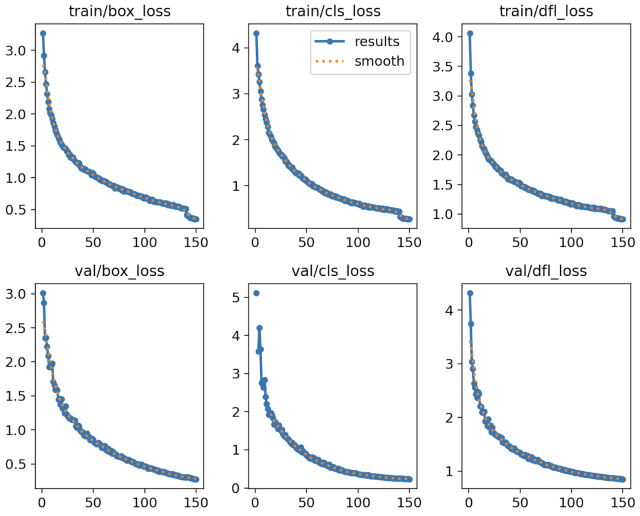

Figure 12. Training and validation losses for YOLO (box, cls, dfl) steadily decrease over epochs, indicating improved model performance.

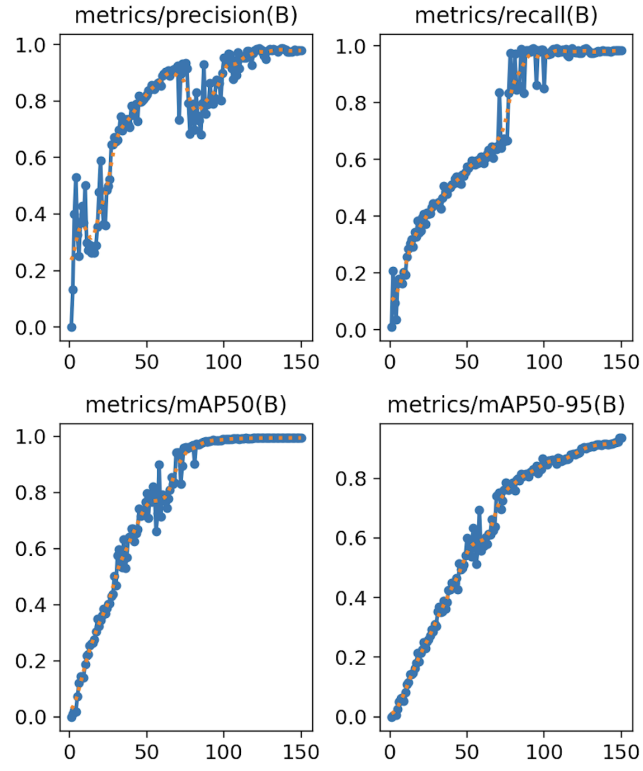

Figure 13. Precision, recall, and mAP for YOLO consistently improve over the course of training, reflecting enhanced object detection accuracy and robustness.

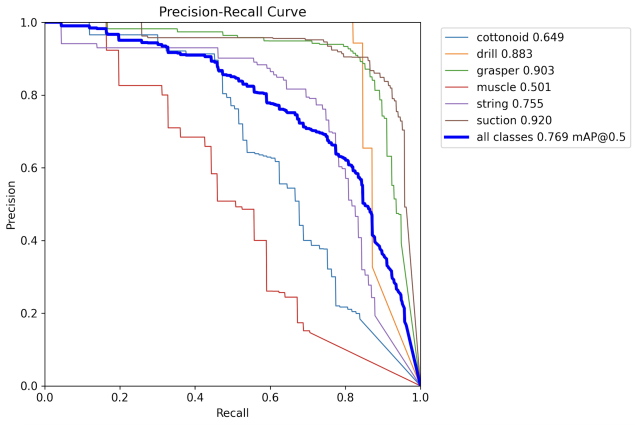

Figure 14. Class-wise Precision-Recall curves for YOLO highlighting detection performance and overall mAP.

inference and detecting small or occluded objects, which are common in surgical environments.

For comparison, other models such as AutoML, RetinaNet, and YOLOv3 achieved mAP scores of 0.708, 0.669, and 0.527, respectively. AutoML, while demonstrating competitive performance, requires extensive computational resources for model selection and optimization. RetinaNet, known for its focus on handling class imbalance using focal loss, performed reasonably well but did not match the efficiency and accuracy of YOLOv8. YOLOv3, an earlier version of the YOLO family, lagged behind the newer YOLOv8 due to its outdated architecture and less advanced feature extraction techniques.

These results collectively underscore the advantages of YOLOv8 as a robust and efficient tool for surgical instrument

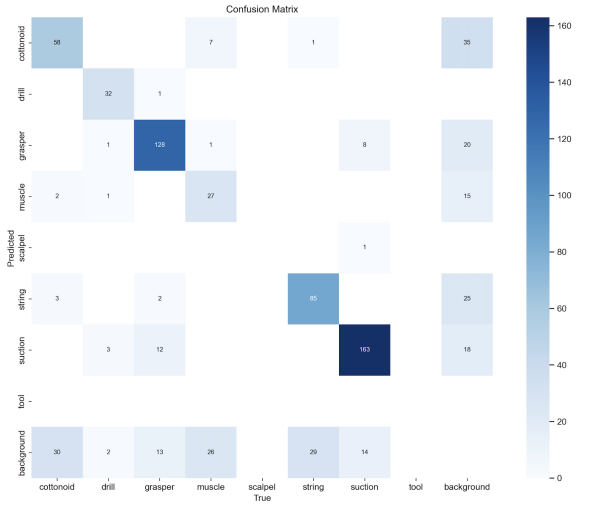

Figure 15. Confusion Matrix (YOLO). The matrix visualizes the classification performance across the eight classes, highlighting areas of strength and misclassification.

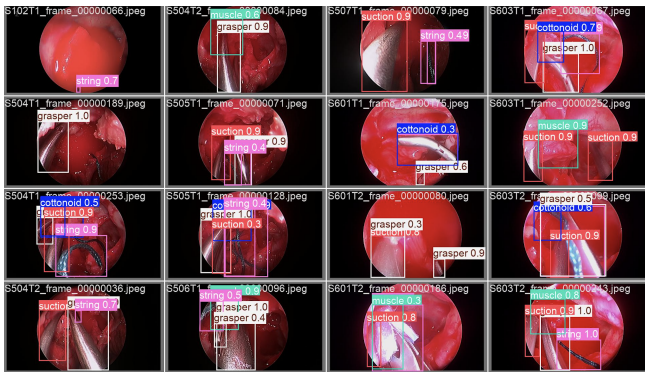

Figure 16. Sample YOLO model predictions on test images.

detection, particularly in real-time applications. Its combination of speed and accuracy makes it a standout choice for dynamic and complex environments like surgical procedures. Additionally, the performance gap between YOLOv8 and other models reaffirms the importance of continually advancing object detection architectures to address the growing demands of real-world applications.

This comparative analysis not only highlights YOLOv8's strengths but also provides a basis for future research to explore enhancements in detection models. Leveraging the lessons learned from these results, future work could focus on integrating the benefits of YOLOv8 with other state-of-the-art models or exploring hybrid approaches to achieve even better performance.

Model	mAP
YOLOv8	0.769
Faster R-CNN	0.625
AutoML	0.708
RetinaNet	0.669
YOLOv3	0.527

Figure 17. Comparative mAP scores of various object detection models.

7 Broader Impacts / Discussion

The integration of real-time object detection models, such as Faster R-CNN and YOLOv8, into surgical workflows represents a transformative step in medical technology. These models can assist in automating repetitive and error-prone tasks, providing surgeons with critical, real-time information about the instruments in use. By enhancing surgical precision and improving tool tracking, this technology has the potential to reduce operation times, minimize errors, and ultimately contribute to better patient outcomes. Furthermore, these advancements align with the broader goal of integrating artificial intelligence into healthcare systems to support decision-making and improve operational efficiency.

One of the significant strengths of this approach lies in its robustness under challenging surgical conditions. The models effectively address issues such as occlusions, overlapping instruments, and variable lighting. This ensures their reliability and accuracy, even in highly dynamic and complex surgical environments. Such robustness is critical in real-world applications, where consistent performance can make the difference between successful and compromised outcomes.

Scalability is another key benefit of these models. The ability to adapt them to detect additional surgical tools or to apply them in other domains, such as robotic-assisted surgeries or medical imaging for anomaly detection, showcases their versatility. This flexibility not only expands their use cases but also underscores their importance in the broader landscape of medical technology.

Looking forward, there are exciting prospects for improving and expanding upon this work. Enhancing the dataset with more diverse and representative samples can address existing biases and improve model generalization across varied surgical scenarios. Additionally, integrating these models

with surgical robots could enable seamless collaboration between AI systems and surgeons, further reducing the cognitive load during operations. Hybrid approaches that combine the precision of Faster R-CNN with the speed of YOLOv8 or advanced architectures like EfficientDet and Vision Transformers (ViT), also present a promising avenue for future research.

Despite these promising implications, the study is not without limitations. Faster R-CNN's slower inference speed makes it less suitable for real-time applications, particularly in high-pressure surgical environments. Similarly, YOLOv8, while faster, faces challenges with detecting small or partially occluded objects. These limitations highlight the need for continued refinement of the models and exploration of new architectures that can balance speed and accuracy more effectively.

In conclusion, the adoption of these advanced object detection models in surgical settings marks a significant advancement in medical AI. By addressing both the immediate and long-term needs of surgical workflows, these models offer a pathway to safer, more efficient, and technology-driven healthcare systems. However, addressing their limitations and broadening their applicability remains crucial for maximizing their impact on the field.

8 Conclusion

In this study, we evaluated the performance of Faster R-CNN and YOLOv8 for detecting surgical instruments, highlighting YOLOv8's superior mean Average Precision (mAP), making it more suitable for dynamic surgical environments. While Faster R-CNN demonstrated good precision, its slower inference speed limits its applicability in real-time tasks. Future work can focus on utilizing the full dataset, which was previously constrained by computational resources, to enhance model accuracy and robustness. Additionally, exploring other detection algorithms such as Vision Transformer and EfficientDet could provide alternative solutions. These advancements, combined with further optimization for complex surgical scenarios, can broaden the applicability of detection models and advance their integration into surgical workflows for safer and more efficient healthcare practices.

9 Work Distribution

Shubham Laxmikant Deshmukh played a pivotal role in managing the transfer learning and training processes for the Faster R-CNN model, leveraging ResNet-50 as the backbone for surgical instrument detection in real-time applications. His responsibilities included converting annotation labels from the YOLO COCO dataset format to the PASCAL VOC format, a necessary step for training the Faster R-CNN model. Upon completing the training, he ensured the trained model, along with the pretrained weights, was optimized for testing and ready for deployment across various devices. Additionally, he explored alternative architectures, such as RetinaNet, EfficientDet, and Detection Transformer, utilizing transfer learning to adapt these models to the custom dataset. However, due to the high computational demands of these models, the available resources were insufficient to meet the training requirements.

Meanwhile, Pradyumna focused on preprocessing the data, transfer learning and training the YOLOv8 model for the same task. Following the independent training of both models, we collaboratively compared their performance on the SOCAL dataset. This comparison aimed to identify the model with the best performance. Based on these results, we explored potential improvements and proposed better solutions for future work, leveraging the insights gained from this comparative analysis.

References

[1] Kugener et al. "Utility of the Simulated Outcomes Following Carotid Artery Laceration Video Data Set for Machine Learning Applications." JAMA Network Open. 2022;5(3):e223177. doi:10.1001/jamanetworkopen.2022.3177

[2] Kugener G, Pangal DJ, Cardinal T, Collet C, Lechtholz-Zey E, Lasky S, Sundaram S, Markarian N, Zhu Y, Roshannai A, Sinha A, Han XY, Papyan V, Hung A, Anandkumar A, Wrobel B, Zada G, Donoho DA. Utility of the Simulated Outcomes Following Carotid Artery Laceration Video Data Set for Machine Learning Applications. JAMA Netw Open. 2022 Mar 1;5(3):e223177. doi: 10.1001/jamanetworkopen.2022.3177. PMID: 35311962; PMCID: PMC8938712.

[3] Y. Wang, Q. Sun, G. Sun, L. Gu and Z. Liu, "Object Detection of Surgical Instruments Based on YOLOv4," 2021 6th IEEE International Conference on Advanced Robotics and Mechatronics (ICARM), Chongqing, China, 2021, pp. 578-581, doi: 10.1109/ICARM52023.2021.9536075.

[4] Zheng, L., Liu, Z. (2023). Real Time Surgical Instrument Object Detection Using YOLOv7. In: Stanimirović, P.S., Zhang, Y., Xiao, D., Cao, X. (eds) 6th EAI International Conference on Robotic Sensor Networks. ROSENET 2022. EAI/Springer Innovations in Communication and Computing. Springer, Cham. https://doi-org.ezproxy.lib.vt.edu/10.1007/978-3-031-33826-7

[5] Bamba, Y., Ogawa, S., Itabashi, M. et al. Object and anatomical feature recognition in surgical video images based on a convolutional neural network. Int J CARS 16, 2045–2054 (2021). https://doi-org.ezproxy.lib.vt.edu/10.1007/s11548-021-02434-w

[6] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 779-788, doi: 10.1109/CVPR.2016.91.

[7] T.-Y. Lin, P. Goyal, R. Girshick, K. He and P. Dollár, "Focal Loss for Dense Object Detection," 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 2017, pp. 2999-3007, doi: 10.1109/ICCV.2017.324.

[8] Xin He, Kaiyong Zhao, Xiaowen Chu, AutoML: A survey of the state-of-the-art, Knowledge-Based Systems, Volume 212, 2021, 106622, ISSN 0950-7051, https://doi.org/10.1016/j.knosys.2020.106622.

[9] Unadkat V, Pangal DJ, Kugener G, Roshannai A, Chan J, Zhu Y, Markarian N, Zada G, Donoho DA. Code-free machine learning for object detection in surgical video: a benchmarking, feasibility, and cost study. Neurosurg Focus. 2022 Apr;52(4):E11. doi: 10.3171/2022.1.FOCUS21652. PMID: 35364576.

[10] F. Zhuang et al., "A Comprehensive Survey on Transfer Learning," in Proceedings of the IEEE, vol. 109, no. 1, pp. 43-76, Jan. 2021, doi:

10.1109/JPROC.2020.3004555.

[11] R. Girshick, "Fast R-CNN," 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 2015, pp. 1440-1448, doi: 10.1109/ICCV.2015.169.

[12] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.

[13] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in Advances in Neural Information Processing Systems 28 (NIPS 2015), pp. 91-99.

[14] A. Buslaev, A. Parinov, E. Khvedchenya, V. Iglovikov, and A. Kalinin, "Albumentations: Fast and Flexible Image Augmentations," Information, vol. 11, no. 2, p. 125, 2020, doi:10.3390/info11020125.

[15] Li, J., Wang, X., Duan, Y., Zhang, W. Multi-scale object detection in remote sensing imagery with convolutional neural networks. *ISPRS Journal of Photogrammetry and Remote Sensing*. May 2018;145:3–17. doi:10.1016/j.isprsjprs.2018.04.003.

[16] Gopalakrishnan, N., Rajamohan, V., Narayanan, S., Gopalakrishnan, N. Performance Evaluation of Deep CNN-Based Crack Detection and Localization Techniques for Concrete Structures. *Sensors*. March 2021;21(5):1688. doi:10.3390/s21051688. License: CC BY 4.0.

[17] Jacob Solawetz and Francesco. What is yolov8? the ultimate guide., 2023. 04-30-2023