

Cholecystectomy Surgical Instrument Detection Using Variants of YOLOv8

Muhammad Adil Raja, Róisín Loughran, Fergal McCaffery

Regulated Software Research Center (RSRC)

Dundalk Institute of Technology (DkIT)

Dundalk, Ireland

adil.raja, roisin.loughran, fergal.mccaffery@dkit.ie

Abstract—Advancements in object detection technologies have ushered in a new era of precision and efficiency in Computer Aided Laparoscopy (CAL), offering transformative benefits to surgical interventions. The importance of adopting and integrating state-of-the-art object detection methodologies within CAL systems is quite critical. Modern object detection algorithms enable CAL systems to accurately localize and track surgical instruments in real-time laparoscopic video feeds. These advancements empower surgeons with enhanced visualization, improved navigation within anatomical structures, and precise instrument guidance during procedures. Furthermore, the seamless integration of object detection technologies with CAL facilitates automated tool recognition, reduces surgical errors, and enhances surgical outcomes. By harnessing the capabilities of deep learning, Convolutional Neural Networks (CNNs), and computer vision techniques, CAL systems can optimize surgical workflow, increase patient safety, and foster a new paradigm of data-driven surgical precision.

Recently version 8 of the famous You Look Only Once (YOLO) algorithm was released. Like all the past releases, it has been claimed that this version is better at detection accuracy as well as computational efficiency. This paper examines the performance of YOLOv8 at tool detection in a CAL context. We employed a well-known laparoscopy tool detection benchmark dataset in this research. Models with superior performance have been obtained as a result of this research. Models are superior in terms of both detection accuracy as well as inference speed. Moreover, the models are ready to be deployed into a production environment. The results that are reported in this paper are not only useful for the surgical community but also for the benchmarking of the YOLO algorithm.

Keywords—Computer-aided laparoscopy, cholecystectomy, object detection, deep learning, convolutional neural networks.

I. INTRODUCTION

The role Computer Vision (CV) is playing in Computer Aided Laparoscopy (CAL) is quite commendable. Remote robotic surgery is now a reality through which a surgeon can perform procedures overseas. In comparison with traditional surgery, minimal incisions, lesser blood loss and shorter recovery times are hallmarks of CAL [1]. In the future, robotic surgery technology can make significant contributions to an already over-burdened enterprise. An order of hundreds of millions of surgeries are performed annually worldwide [2]. However, there is a backlog of the same order of magnitude

of surgeries [2]. The existing supply and availability of human surgeons can clearly not catch up with the demand for surgical operations. Robot-assisted surgery can significantly share this burden with the surgeons.

Surgery is a delicate art that requires precision and patience. In order for the robotic apparatus for surgery to be adequate, it has to make accurate predictions about surgical tools in real-time. This puts onerous expectations on the Machine Learning (ML) community to improve their object detection algorithms. Object detection, in turn, has also become more accurate and faster over the years. Although advancements in High Performance Computing (HPC) and Graphical Processing Unit (GPU) hardware have helped the capabilities of object detection markedly, further improvements can always be made.

Recently the eighth version of the You Look Only Once (YOLO) family of algorithms was released [3]. The new algorithm was developed by Ultralytics, an enterprise that is responsible for developing some of the earlier versions of the algorithm too. The new version touts better prediction accuracy as well as faster speed for object detection tasks. In this paper, we report the performance of this algorithm on tool detection tasks in CAL settings. In our work, we employed an open-source database of CAL images.

The rest of this paper is organized as follows: In section II, we provide a brief background of CAL. Section III is dedicated to an introduction to the YOLO algorithm. Section IV presents our experimental details. In section V, we have reported the results of our study. Finally, section VI concludes this paper.

II. COMPUTER AIDED SURGERY AND LAPAROSCOPY

CAL technologies can help alleviate some of the burden on surgeons. Surgery is a delicate art that has to be conducted with extreme caution. Given the intricacies involved in the operation, it is not likely that robotics will completely replace humans ever. However, as a helping tool, CAL can enormously affect the efficiency of surgical operations.

One of the important domains in CAL is CV. Object detection based on CV has tremendously helped to achieve sophistication in CAL. What was considered a formidable task a couple of decades ago, has now become a reality due to advancements in HPC as well as ML algorithms. The advent of Convolutional Neural Networks (CNNs) and GPU hardware has made it possible to tag surgical tools in real-time

This research was funded by the Technological University Transfer Fund (TUTF) of the Higher Education Authority (HEA) and DKIT.

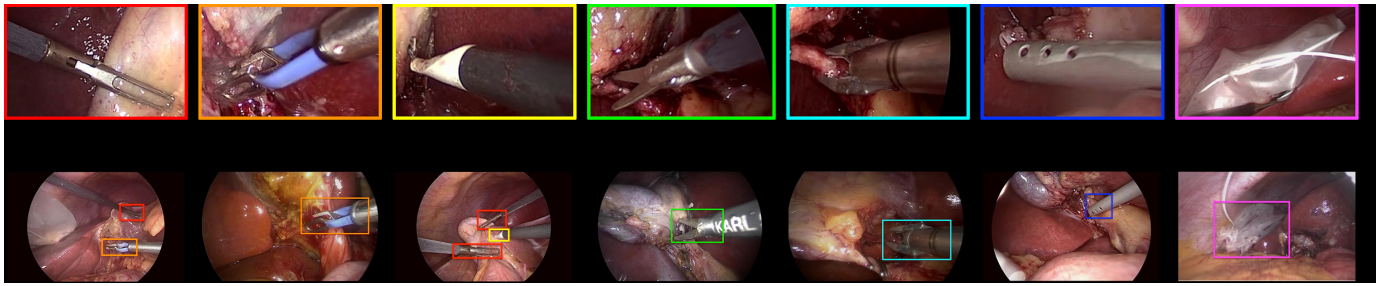


Fig. 1. Examples of CAL tools in different contexts during surgery. The top row shows tools and the bottom row shows bounding boxes around those tools.

with a high degree of accuracy. Progress in CNNs has been gradual over the years ever since their mainstream inception as a practical tool for CV [4]. There have been numerous successful applications of CNNs for CAL tool detection in recent years [5]–[7].

III. OBJECT DETECTION AND THE YOLO ALGORITHM

Object detection is the task of identifying objects in images and tagging them with appropriate labels. To this end, this task is more intricate than image classification. In image classification, images are classified according to the types of objects they contain. Normally the number of types of objects is restricted to one. For instance, if an image has a picture of a cat or a dog inside it, it is classified as a cat or a dog respectively. However, the exact location of the object in the image is not known. Nowadays this is considered insufficient information as normally practitioners also want to know the exact location of the object(s) in the image. Hence, this gives rise to the need for object detection. Object detection has come a long way since the practical implementation of deep learning and CNNs. A comprehensive account of object detection is beyond the scope of this paper and can be read elsewhere [8]. For the purpose of brevity, we shall give a short overview of the YOLO algorithm. Unlike the traditional object detection algorithms, YOLO took a stride in object detection by posing this problem as a regression problem as opposed to a classification problem, as it would be done by the erstwhile algorithms. Thus, what the algorithm does is that it divides an image into a number of grid cells. It then performs a regression operation between each of the bounding boxes of each grid cell with the coordinates of the target image's bounding box(es). The bounding box that overlaps the most with the target bounding box is chosen to locate the target. After this, the algorithm computes a classification task to assay the label of the image in the bounding box. This is a very simplistic explanation of the algorithm; the algorithm in itself is quite complicated, involving much more mathematical detail. Detailed explanations of the YOLO algorithm and its variants can be read elsewhere [9], [10]. Nonetheless, most variants of the YOLO algorithm differ in the architecture of the CNN [10]. The main functioning is based on this simple philosophy i.e., bounding box regression followed by classification of what lies within the regressed bounding box. The key idea that

distinguishes this algorithm from the rest is that it does both bounding box regression and object classification in a single shot. Both regression and classification happen simultaneously.

YOLOv8 has five variants depending on the size of the model. In order of increasing size, these are YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, and YOLOv8x. These are nano, small, medium, large, and extra large.

In recent years, the YOLO algorithm has also been successfully applied for CAL tool detection [11]–[16]. All of these studies have employed a much earlier version of the YOLO algorithm.

In our study, we have employed all the variants of YOLOv8 model to achieve the reported results.

IV. EXPERIMENTS

We employed the well-known m2cai16-tool-locations dataset [17] in our research. This dataset contains spatial tool annotations for 2,811 frames involving 10 videos. The dataset consists of 3,141 annotations of 7 classes of surgical instruments. Names of these classes are grasper, bipolar, hook, scissors, clipper, irrigator, and specimen bag. Fig. 1 shows example pictures of each one of these. The top row shows pictures of each of these tools, whereas the bottom row shows bounding boxes for each of the tools ¹.

There is an average of 1.2 labels per frame. The dataset is divided into 1405, 843, and 563 instances for training, validation during training, and testing on unseen data respectively. This roughly forms splits of 50%, 30%, and 20% of the whole data respectively. Table I shows the overall distribution of the data. More specifically, it shows how many occurrences of each instrument were present in training, validation, and test datasets.

We trained all the five variants of the YOLOv8 algorithm.

We conducted the training of s, m, l, and x models on the Kay supercomputer provided by the Irish Center for High end Computing (ICHEC). The computing node on which training was carried out has two NVIDIA Tesla V100 16GB PCIe (Volta architecture) GPUs. Both GPUs have 5,120 CUDA cores and 640 Tensor Cores. Training of the smallest model (i.e. YOLOv8n) was conducted using an NVIDIA GeForce

¹This figure was taken from the website of the data repository: <https://ai.stanford.edu/~syueung/tool detection.html>

TABLE I
THE NUMBER OF INSTANCES OF VARIOUS TOOLS IN THE DATASET

Instrument	Training	Validation	Test
Grasper	707	420	293
Bipolar	215	120	95
Hook	165	79	64
Scissors	197	107	84
Clipper	220	116	64
Irrigator	228	173	84
Specimen Bag	241	138	96

RTX 2060 with Max-Q Design, having 5927MiB of on-chip memory. Inference results for all the models were carried out using this latter GPU.

Default values for the hyperparameters for training were used except for the fact that the batch size was set to 16 and the number of training epochs was 200 in accordance with [18]. Since the image sizes in the dataset are 596×334 pixels, the image size for training, validation, and inference was set to 596 pixels, as opposed to the default of 640. Since the YOLO package, could not deal with this size, it automatically re-adjusted the size to 608 pixels. This is an inherent feature of the YOLO repository provided by Ultralytics.

A. Evaluation Metrics

We used the commonly used evaluation metrics that are ubiquitous in the CV literature. YOLOv8 makes use of these metrics. Among these are Precision (P), Recall (R), mean Average Precision (mAP)50, and mAP50-95. The definition and formulae for these metrics are not given here for the purpose of brevity. However, these descriptions can be found in existing literature [18].

V. RESULTS

In this section, we report the results of our training of all the five variants of YOLOv8 models. The results are related to object detection of surgical tools used in CAL of cholecystectomy. Fig. 2 shows the confusion matrix related to the various variants of the trained models. The results are reported in terms of percentages.

Fig. 3 shows pictorial results for the curious reader. The three pictures in the left column show the ground truth labels of different batches of data. Whereas the pictures in the right column show the predictions of the trained model. These figures are related to the unseen test data.

Finally, Fig. 4 shows various important plots related to the processing of the unseen test data by the extra large model (YOLOv8x). The plots contain curves related to each of the tools alone as well as the combined data Fig. 4a, 4b, and 4c show the F1, precision, and recall confidence curves. More specifically, these curves provide us with the maximum values of these metrics for given confidence levels. F1 score is the weighted average of precision and recall. Recall tells us what percentage of the actual positives was predicted correctly. Fig. V shows a precision-recall curve. This is an important curve in classification problems that tells us about the tradeoff between these two important metrics.

Table II shows the overall performance of YOLOv8x in terms of different performance metrics. Results are shown for all classes of instruments. These results correspond with the dataset used for testing on the unseen data.

TABLE II
PERFORMANCE OF YOLOV8X ON UNSEEN TEST DATA

Class	Instances	P	R	mAP50	mAP50-95
all	780	0.925	0.91	0.943	0.59
Grasper	293	0.887	0.828	0.897	0.521
Bipolar	95	0.954	0.947	0.969	0.573
Hook	64	0.96	0.984	0.986	0.718
Scissors	84	0.906	0.918	0.949	0.548
Clipper	64	0.98	0.844	0.904	0.645
Irrigator	84	0.905	0.911	0.937	0.48
Specimen Bag	96	0.882	0.935	0.959	0.643

Table III shows a comparison of different models in terms of mAP50 metric. Again, the results are related to the unseen test data.

TABLE III
COMPARISON OF DIFFERENT MODELS IN TERMS OF MAP50

Class	YOLOv8n	v8s	v8m	v8l	v8x
All tools	0.959	0.939	0.94	0.953	0.943
Grasper	0.915	0.874	0.886	0.899	0.897
Bipolar	0.97	0.955	0.957	0.976	0.969
Hook	0.995	0.964	0.972	0.979	0.986
Scissors	0.972	0.977	0.948	0.957	0.949
Clipper	0.961	0.913	0.923	0.952	0.904
Irrigator	0.942	0.939	0.946	0.951	0.937
SpecimenBag	0.956	0.952	0.947	0.959	0.959

Table IV shows the speed of different variants of the YOLOv8 models per image in milliseconds (ms) at various stages of inference. It can be seen that the most amount of time by each model is in the inference stage. The rest of the stages consume almost the same amount of time. Notably, YOLOv8n has the best inference speed, whereas YOLOv8x consumes the most amount of time.

VI. CONCLUSIONS

In this paper, we have presented results related to tool detection of CAL instruments. The well-known m2cai16-tool-locations cholecystectomy dataset was used to train all the variants of YOLOv8. Seven different types of instruments were present in the data. Results of the trained object detection models have been reported in this paper, which compare favorably with other experiments of this kind.

TABLE IV
INFERENTIAL SPEED OF DIFFERENT MODELS PER IMAGE

Model	Pre-processing	Inference	Loss	Post-Processing
YOLOv8n	0.2 ms	2.5 ms	0.0 ms	1.0 ms
YOLOv8s	0.2	4.6	0.0	1.7
YOLOv8m	0.2	10.7	0.0	0.7
YOLOv8l	0.3	17.7	0.0	0.7
YOLOv8x	0.2	25.6	0.0	0.7

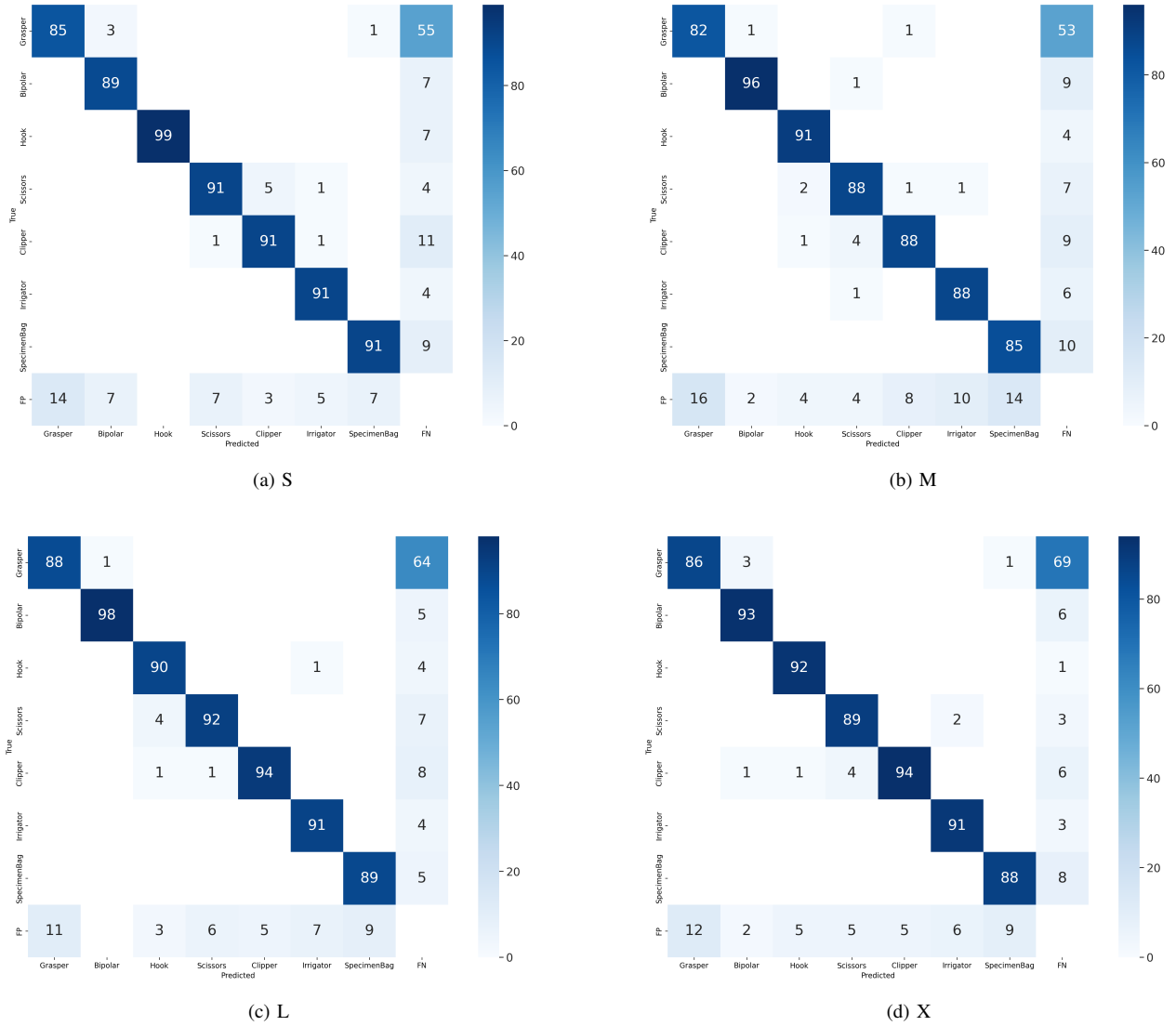


Fig. 2. Confusion matrices for different models on the unseen test data.

This research was a part of our wider efforts to develop efficient and effective algorithms for surgical tool detection. At the time of conducting this research, [11] was the most relevant research involving the use of YOLOv4 for surgical tool detection. By the time this research was completed, [18] emerged online as new research involving YOLOv8 for the same problem, displaying the speed at which this technology is advancing. The research and results produced here are still novel and original as we have employed the full gamut of variants of YOLOv8 in this research.

REFERENCES

- [1] G. Ietto, F. Amico, G. Pettinato, V. Iori, and G. Carcano, "Laparoscopy in emergency: why not? advantages of laparoscopy in major emergency: a review," *Life*, vol. 11, no. 9, p. 917, 2021.
- [2] J. Meara, A. Leather, L. Hagander, B. Alkire, N. Alonso, E. Ameh *et al.*, "Global surgery 2030: evidence and solutions for achieving health, welfare, and economic development. lancet.[internet]. 2015 aug [cited jan 18, 2018]; 386 (9993): 569-624."
- [3] G. Jocher, A. Chaurasia, and J. Qiu, "Yolo by ultralytics," *URL: <https://github.com/ultralytics/ultralytics>*, 2023.
- [4] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai *et al.*, "Recent advances in convolutional neural networks," *Pattern recognition*, vol. 77, pp. 354–377, 2018.
- [5] M. K. Hasan, L. Calvet, N. Rabbani, and A. Bartoli, "Detection, segmentation, and 3d pose estimation of surgical tools using convolutional neural networks and algebraic geometry," *Medical Image Analysis*, vol. 70, p. 101994, 2021.
- [6] H. Saeidi, J. D. Opfermann, M. Kam, S. Wei, S. Léonard, M. H. Hsieh, J. U. Kang, and A. Krieger, "Autonomous robotic laparoscopic surgery for intestinal anastomosis," *Science robotics*, vol. 7, no. 62, p. eabj2908, 2022.
- [7] P. Mascagni, D. Alapatt, L. Sestini, M. S. Altieri, A. Madani, Y. Watanabe, A. Alseidi, J. A. Redan, S. Alfieri, G. Costamagna *et al.*, "Computer vision in surgery: from potential to clinical value," *npj Digital Medicine*, vol. 5, no. 1, p. 163, 2022.
- [8] K. Ragland and P. Tharcis, "A survey on object detection, classification and tracking methods," *Int. J. Eng. Res. Technol.*, vol. 3, no. 11, pp. 622–628, 2014.
- [9] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE*

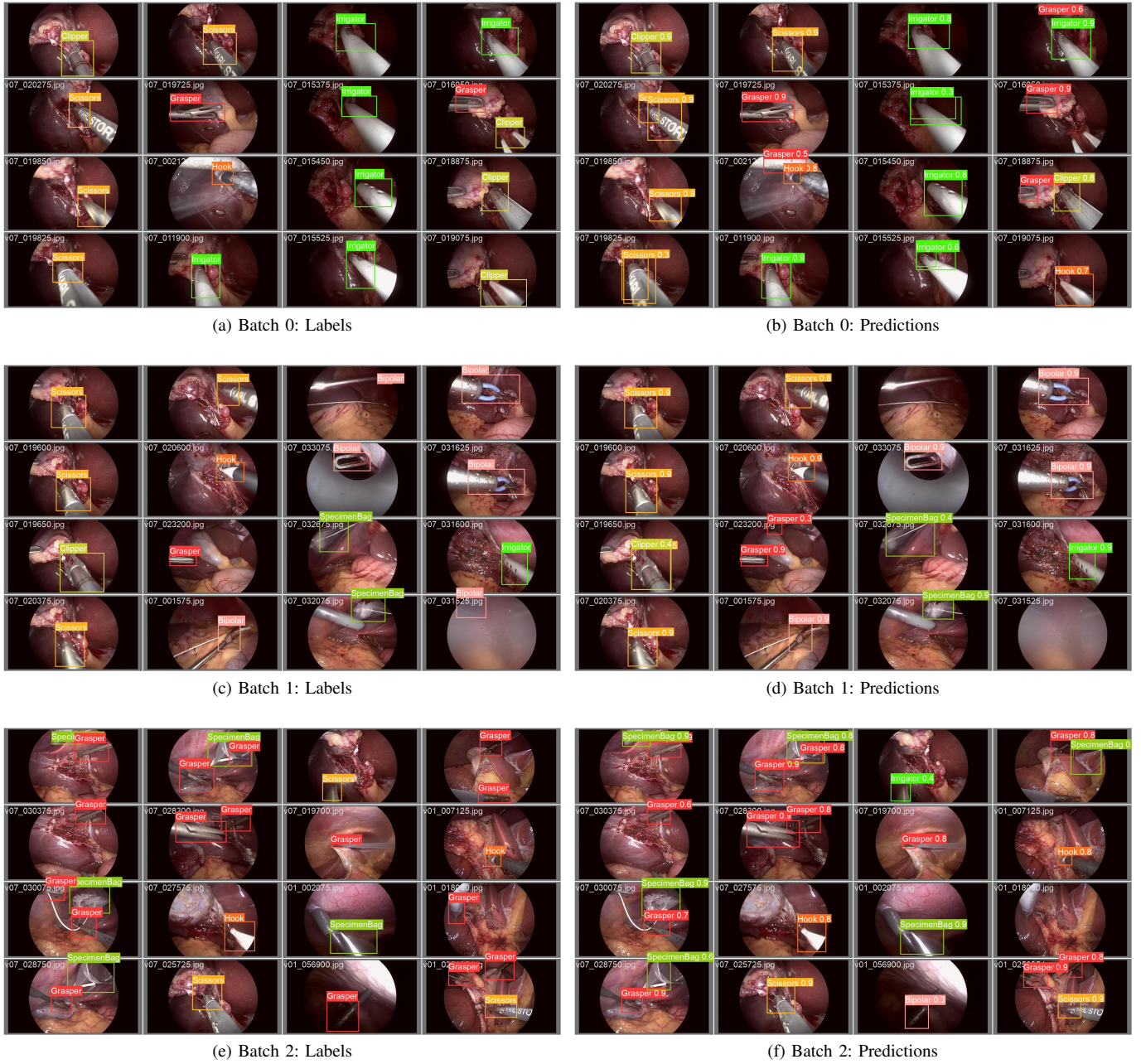


Fig. 3. Ground truth labels for various batches of validation data versus the predicted labels. Different types of instruments can be distinguished on the basis of the color of the bounding box. Grasper = Green. Bipolar = Pink. Hook = Dark Orange. Scissors = Mustard. Clipper = Neon. Irrigator = Green. Specimen Bag = Parrot green.

- conference on computer vision and pattern recognition, 2016, pp. 779–788.
- [10] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, “A review of yolo algorithm developments,” *Procedia Computer Science*, vol. 199, pp. 1066–1073, 2022.
- [11] Y. Wang, Q. Sun, G. Sun, L. Gu, and Z. Liu, “Object detection of surgical instruments based on yolov4,” in *2021 6th IEEE International Conference on Advanced Robotics and Mechatronics (ICARM)*. IEEE, 2021, pp. 578–581.
- [12] K. M. Ali, M. Saruwatari, K. Jawed, Y. Kim, S. Park, S. Cha, B. Ning, and R. J. Cha, “Artificial intelligence-assisted laparoscopic cholecystectomy in a preclinical swine model,” in *2023 57th Annual Conference on Information Sciences and Systems (CISS)*. IEEE, 2023, pp. 1–4.
- [13] C. Yang, Z. Zhao, and S. Hu, “Image-based laparoscopic tool detection and tracking using convolutional neural networks: a review of the literature,” *Computer Assisted Surgery*, vol. 25, no. 1, pp. 15–28, 2020.
- [14] J. Peng, Q. Chen, L. Kang, H. Jie, and Y. Han, “Autonomous recognition of multiple surgical instruments tips based on arrow obb-yolo network,” *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–13, 2022.
- [15] B. Choi, K. Jo, S. Choi, and J. Choi, “Surgical-tools detection based on convolutional neural network in laparoscopic robot-assisted surgery,” in *2017 39th annual international conference of the IEEE engineering in medicine and biology society (EMBC)*. Ieee, 2017, pp. 1756–1759.
- [16] Y. Yamazaki, S. Kanaji, T. Matsuda, T. Oshikiri, T. Nakamura, S. Suzuki, Y. Hiasa, Y. Otake, Y. Sato, and Y. Kakeji, “Automated surgical

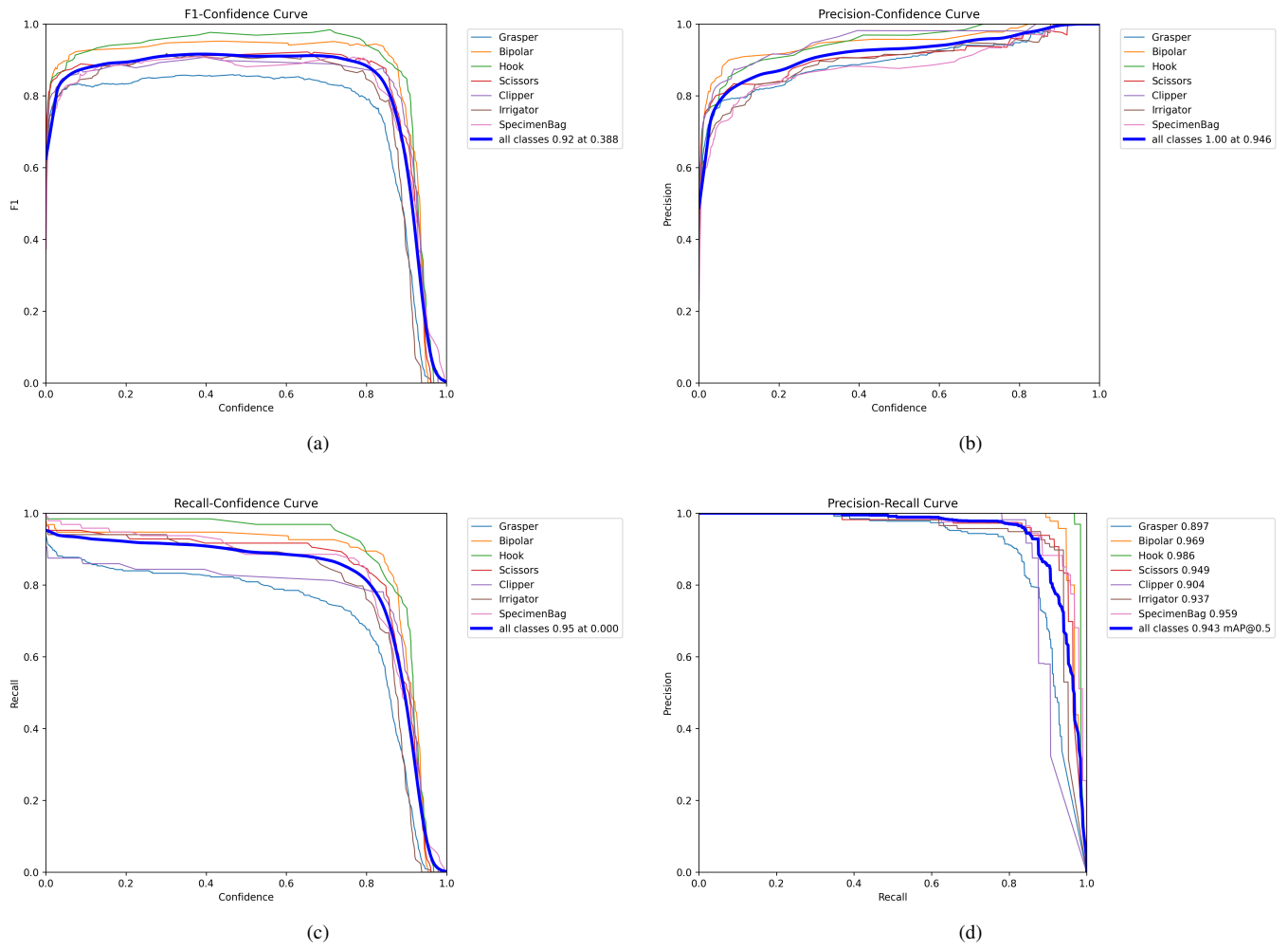


Fig. 4. Various plots related to the unseen test data.

instrument detection from laparoscopic gastrectomy video images using an open source convolutional neural network platform,” *Journal of the American College of Surgeons*, vol. 230, no. 5, pp. 725–732, 2020.

- [17] A. Jin, S. Yeung, J. Jopling, J. Krause, D. Azagury, A. Milstein, and L. Fei-Fei, “Tool detection and operative skill assessment in surgical videos using region-based convolutional neural networks,” *IEEE Winter Conference on Applications of Computer Vision*, 2018.
- [18] H.-B. Le, T. D. Kim, M.-H. Ha, A. L. Q. Tran, D.-T. Nguyen, and X.-M. Dinh, “Robust surgical tool detection in laparoscopic surgery using yolov8 model,” in *2023 International Conference on System Science and Engineering (ICSSE)*. IEEE, 2023, pp. 537–542.