A comparative analysis of multi-backbone Mask R-CNN for surgical tools detection

Gioele Ciaparrone
Neuronelab, DISA-MIS
Università degli Studi di Salerno
Salerno, Italy
gciaparrone@unisa.it

Francesco Bardozzo
Neuronelab, DISA-MIS
Università degli Studi di Salerno
Salerno, Italy
fbardozzo@unisa.it

Mattia Delli Priscoli
Neuronelab, DISA-MIS
Università degli Studi di Salerno
Salerno, Italy
mdellipriscoli@unisa.it

Juanita Londoño Kallewaard
Neuronelab, DISA-MIS
Università degli Studi di Salerno
Salerno, Italy
Faculty of Engineering - UTP
Pereira, Colombia
j.londonokallewaa@studenti.unisa.it

Maycol Ruiz Zuluaga
Neuronelab, DISA-MIS
Università degli Studi di Salerno
Salerno, Italy
Faculty of Engineering - UTP
Pereira, Colombia
m.ruizzuluaga@studenti.unisa.it

Roberto Tagliaferri
Neuronelab, DISA-MIS
Università degli Studi di Salerno
Salerno, Italy
robtag@unisa.it

***Abstract*—Real-time surgical tool segmentation and tracking based on convolutional neural networks (CNN) has gained increasing interest in the field of mini-invasive surgery. In fact, the application of this novel artificial vision technologies allows both to reduce surgical risks and to increase patient safety. Moreover, these types of models can be used both to track the tools and detect markers or external artefacts in a real-time video stream. Multiple object detection and instance segmentation can be addressed efficiently by leveraging region-based CNN models. Thus, this work provides a comparison among *state-of-the-art* multi-backbone Mask R-CNNs to solve these tasks. Moreover, we show that such models can serve as a basis for tracking algorithms. The models were trained and tested with a data-set of 4955 manually annotated images, validated by 3 experts in the field. We tested 12 different combinations of CNN backbones and training hyperparameters. The results show that it is possible to employ a modern CNN to tackle the surgical tool detection problem, with the best-performing Mask R-CNN configuration achieving 87% Average Precision (AP) at Intersection over Union (IOU) 0.5.**

I. Introduction

Real-time surgical tool segmentation based on CNNs has gained increasing interest in the field of mini-invasive surgery (MIS). MIS is adopted to reduce patient pain and post-operative complications. However, due to limited operating space and insufficient feedback, the risk to cause damage to surfaces and internal organs is concrete. For this reason, considering the impressive results previously achieved with the use of deep learning in various tasks in the biomedical field [1]–[3], surgical tool detection and tracking based on CNN are applied to provide an augmented perception to the surgeon, thereby reducing surgical risks and preserving patient safety. Therefore, real-time tool detection is an essential component to avoid operative and post-operative complications [4]. First, advanced computer vision approaches could be useful to detect external objects, such as surgical markers, and they can help solving the problems of identifying and localizing artefacts of different nature [5]–[12]. Second, hiding objects from the background by segmenting multiple objects is an important task in diagnostics and image-based investigations. In fact, if the region of interest is partially occluded by the presence of surgical instruments, it is possible to remove these tools from the frames. For example, this is particularly useful in two types of registration, which are volume-to-surface or surface-to-surface alignment problems [13], and in 3D soft tissue reconstruction tasks [14]. Third, the segmentation of multiple objects could provide proximity measurements between the left and right robotic tools or between them and the background, by predicting their overlap and possible collisions [15]. To address these tasks, even if several CNN-based models (e.g. U-Net [16]) have been adopted, as far as we know, a comparative study on Mask R-CNN [17] models has never been conducted in this area of medical imaging. While models like U-Net perform semantic image segmentation, that is they are able to segment foreground and to classify it into different classes, the main advantage of employing Mask R-CNN-like models relies in their ability to perform object instance segmentation, that is identifying separate object instances of a target class and predicting their segmentation mask separately; so, while U-Net would only be able to detect pixels belonging to the "surgical tools" class, Mask R-CNN is also able to distinguish among different instances of such tools. In turn, this can be useful as a basis to track each of the tools.

The main contribution of this work is a comparison among different configurations of the state-of-the-art Mask R-CNN detectors for recognizing and segmenting endoscopic surgical tools. Moreover, we evaluate the robustness of these models under challenging conditions, such as low-resolution videos.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

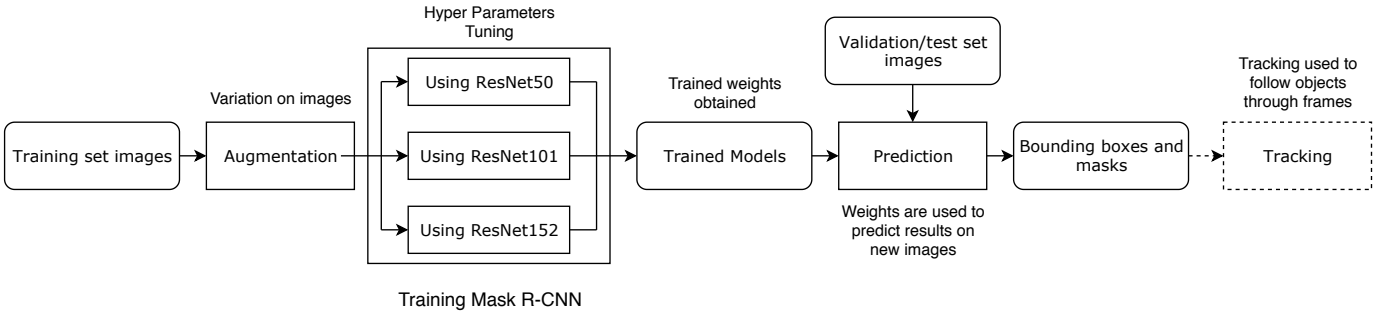


Fig. 1: Pipeline of the implemented system, which consist of two main parts: model training phase, and prediction phase. The bounding boxes and masks output by the Mask R-CNN model were also used as input to a simple tracking algorithm to evaluate the potential of implementing a full tracking pipeline using this model.

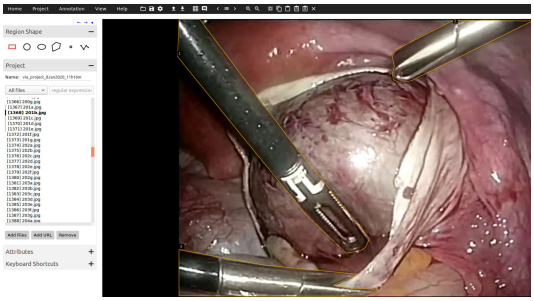
In particular, our work suggests that it is possible to train a high-performance model for detection and segmentation of surgical tools in endoscopic images, and that it can also serve as a base for a tool tracking algorithm.

The organization of this paper is as follows. In Section II the dataset collection, pre-processing and hand curated image labelling procedures are described. Section III, after a description of the employed model, is divided into two sub-sections: sub-section III-A discusses the training procedure, while sub-section III-B describes the data augmentation procedure. In Section IV the experimental results are discussed and a simple tracking algorithm is presented to evaluate the potential of using the model as a base for a tool tracking pipeline. Finally, Section V sums up the results of this study and possible future directions of research.

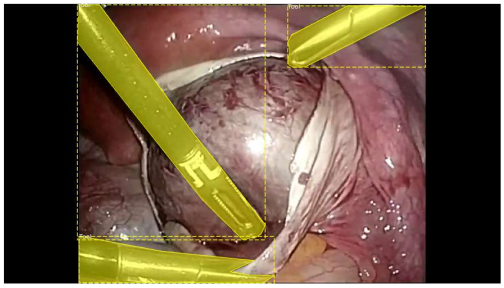
II. Dataset

To build our dataset, we obtained a total of 4198 selected video frames with a resolution of 1920×1080 pixels, from 13 high-quality endoscopic/laparoscopic videos, plus an additional 757 frames from a low-resolution video (384×192 pixels). In particular, some of them contain noise in the form of superimposed written text or minimal graphical user interfaces. We chose to keep those noisy frames, as they can be useful to analyze if the models are able to generalize on unseen conditions and unfiltered noisy frames.

The dataset is divided into three parts: 3195 images for the training set and 290 images for the validation set were extracted from different sections of the same videos. To further test the generalization capabilities of the model, 713 frames from an unseen video have also been extracted and used as a test set. Finally, a low-resolution dataset (757 images) was used as a second test set to evaluate the model robustness to drastic changes in resolution. The images were manually annotated by marking each visible tool with a binary mask [18]–[20]. The annotation procedure was performed using VIA (VGG Image Annotator) [21], [22], and validated by 3 experts in the field. Figure 2 shows the annotation process of a single sample frame. The annotation process involved manually drawing the polygons delimiting each of the tools in every video frame. The class label "Tool" was assigned to every polygon, which represented a foreground mask during the training process. The rest of the image was considered background.



(a) Annotation process of a sample frame.



(b) Foreground annotation sample.

Fig. 2: Manual annotation of a example sample from the endoscopic dataset (2a) and its resulting annotated mask (2b). The process was performed by using VIA (VGG Image Annotator) [21].

III. Models and Methods

Since deep neural networks are the most effective techniques currently available to solve object detection and instance segmentation tasks [23]–[27], we chose to employ the Mask R-CNN architecture [17] to detect and segment surgical tools. The Mask R-CNN model is an evolution of Faster R-CNN [28]. In addition to predicting the bounding boxes containing each instance of the target class(es) and a confidence score for each box, it can also compute a segmentation mask

for each object instance. The Mask R-CNN architecture can be implemented using various backbone network structures, with varying number of layers and complexity. For this study, we chose to use three different backbone architectures: ResNet-50, ResNet-101 and ResNet-152 [29], where 50, 101 and 152, respectively, indicate the number of convolutional layers in the network. More information about the ResNet backbones (such as kernel and output sizes) can be found in the original ResNet article [29].

A. Training procedure

For the training procedure, we exploited transfer learning [30] by initializing the network with pre-trained weights that were obtained on the COCO dataset [31]. The original classification head was replaced by a 2-class classification head (background vs. surgical tools). To evaluate the best model training procedure and backbone architecture, we trained the models by varying different networks and hyperparameters: i) the backbone network employed, ii) regularization parameters, iii) number of epochs, iv) use of data augmentation. In particular, as already mentioned, we used ResNet-50, ResNet-101 and ResNet-152 as backbone networks. Training was performed using Stochastic Gradient Descent (SGD) with learning rate 0.001 and momentum 0.9. We tested two values for the L2-regularization parameter [32] (0.0001, weaker regularization, and 0.001, stronger regularization) and for the number of epochs (25 or 30) [33]. The different experimental setting combinations are shown in Table II, where **Exp** is the *Experiment number*, **Bb** is the *number of layers* of the ResNet backbone, **Reg** is the *regularization parameter*, **Ep** is the *number of epochs*, **Aug** is the *number of augmented images* generated for each original training image (see Section III-B), **PC Spec** is the *hardware* used for each training/test procedure (see Section IV).

The model ability to generalize, the accuracy and performance analyses are assessed on the validation set and tested on the two test videos. The goodness of the model is evaluated using the well-known Average Precision metric (**AP**), as it is described in Section IV.

B. Data augmentation

The segmentation accuracy has been improved by applying data augmentation [34], a well-known technique adopted in real-world problems to improve models accuracy and reduce overfitting to the training set by presenting variations of the same image to the network during training. This helps the model to generalize unseen images, thereby improving its performance on external data [35]. In this work, the following augmentation techniques have been applied: i) rotation ii) scaling iii) flipping iv) perspective changes and v) linear contrast changes. More details about the augmentation parameters are provided in Table I. The efficiency of data augmentation is proved by training the models with different backbones both with and without data augmentation. In Table II, a value of **Aug** $= 0$ indicates the absence of data augmentation, while a value of **Aug** $= 2$ means that the training was performed by adding to the dataset 2 augmented images for each original image in the training set, effectively growing the number of training images from 3195 to 9585. Each augmented image was generated by sequentially applying the previously described augmentation techniques, randomly selecting a value for each transformation parameter listed in Table I. As we will see, in our endoscopic/laparoscopic image dataset, data augmentation turns out to be relevant for an improved detection and segmentation accuracy for our best performing model.

TABLE I: Data augumentation parameters

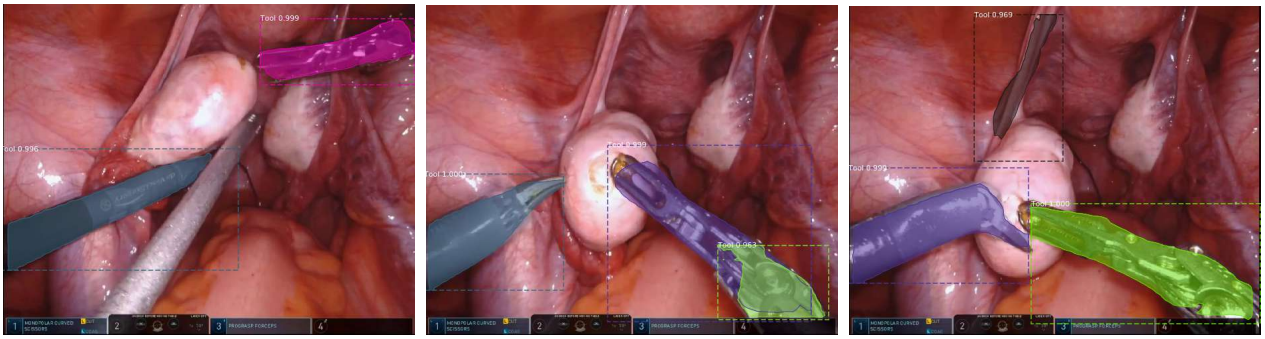
Technique	**Details**
Rotation	10° clockwise and counterclockwise
Scaling	Range [0.8, 1.2]
Flipping	Vertical and horizontal
Perspective change	Range [0.01, 0.1]
Linear contrast change	Range [0.8, 1.2]

IV. Experiments and results

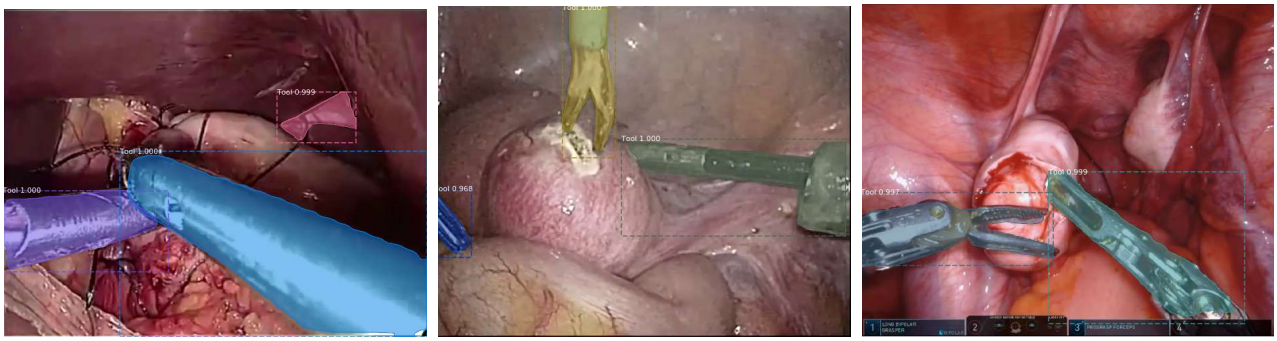
As explained in the previous section, a set of experiments were performed using three versions of Mask R-CNN, trained with different hyperparameters (see Table II). Furthermore, after the train and validation sets, two independent test sets were used. The former has high-resolution images (1920×1080px), but presents visual artefacts, specifically a navigation bar under the images. The latter has no virtual artefacts (such as surgical machine logos, medical notes, virtual markers) but is made of low-resolution images (384×192px). This was done to test the robustness and generalization capabilities of the trained models.

TABLE II: Hyperparameter sets

Exp	**Bb**	**Reg**	**Ep**	**Aug**	**PC Spec**		
					Train	Val	Test
1	101	L2 0.0001	25	0	1	1	2
2	101	L2 0.0001	25	2	1	1	2
3	101	L2 0.0001	30	2	1	1	2
4	101	L2 0.001	30	2	2	1	2
5	50	L2 0.0001	25	0	1	1	2
6	50	L2 0.0001	25	2	1	1	2
7	50	L2 0.0001	30	2	2	1	2
8	50	L2 0.001	30	2	2	1	2
9	152	L2 0.0001	25	0	2	1	2
10	152	L2 0.0001	25	2	1	1	2
11	152	L2 0.0001	30	2	2	2	2
12	152	L2 0.001	30	2	2	2	2



(a)



(b)

Fig. 3: Example of bad prediction (3a) and good prediction results (3b). The quality of the prediction is determined by the type and conditions of the analyzed image and the quality of the trained model.

A. Implementation details

All the processes and experiments were developed in Python 3.7. The following packages are used to implement Mask R-CNN and for image processing: OpenCV [36], MaskRCNN [17], Tensorflow-GPU [37], Keras [38] and imgaug [39]. The pycocotools package was used for evaluation [31].

All the experiments were performed on machines with two different hardware configurations: 1) CPU Intel Core I7-8700, 16GB of Ram, GPU Nvidia GTX 1060 and 2) CPU Intel Core I7-8700, 16GB of Ram, GPU Nvidia GTX 1070. The **PC Spec** column in table II specifies which machine was used for each experiment. The training and inference times thus varied among experiments, according to the backbone and PC used. The fastest model was ResNet-50, as expected, given its lower number of layers, with an inference time of around 0.4 seconds per image on the fastest machine, while the inference time reached 0.6 second per image when using ResNet-152, when using the 1920×1080 px dataset. While, this is still not enough for real-time performance on high-quality video, it shows encouraging results for its use in real-time applications in the near future on more specialized hardware.

B. Metrics

To evaluate the performance of all the trained models, the **AP** [40] was used, both for the bounding boxes and for the masks, as it is common practice in the object detection and instance segmentation tasks [17], [27], [41]. We followed the COCO evaluation protocol and computed the AP at varying levels of bounding box/mask overlap (IOU). In particular, for each results table, we list six different metrics: $\mathbf{AP}^{bb}_{50}$ is the Average Precision at bounding box IOU threshold 0.50, while $\mathbf{AP}^{bb}_{75}$ is the AP at bounding box IOU threshold 0.75; $\mathbf{AP}^{bb}$ indicates the average AP computed at different IOU threshold levels, from 0.5 to 0.95, increasing in steps of 0.05. $\mathbf{AP}^{mask}_{50}$, $\mathbf{AP}^{mask}_{75}$ and $\mathbf{AP}^{mask}$ work similarly, but using mask IOU instead of bounding box IOU, in order to evaluate the mask network branch accuracy.

C. Results and discussion

The results of the analysis are shown in Table III for the validation set, Table IV for the high-resolution test set, and Table V for the low-resolution test set.

According to the presented results, on all considered datasets, the Mask R-CNN with ResNet101 backbone obtained the best performance in the segmentation of surgical tools in endoscopic/laparoscopic images, reaching 92% AP on both boxes and masks at IOU threshold 0.50 on the validation set, and 87% on the high-resolution test set. The ResNet152 backbone also presented good performance on the high-resolution test set, reaching 86% AP at IOU threshold 0.50 with both boxes and masks. We also notice that the AP on boxes and masks are usually highly correlated, showing that whenever a box is correctly identified on a given tool, the mask is also often correctly predicted. As expected, the AP is lower at higher IOU threshold, but still relatively good, reaching,

TABLE III: RESULTS ON THE VALIDATION SET

Experiment	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{mask}	AP^{mask}_{50}	AP^{mask}_{75}
(1) RN101	0.63	0.90	0.72	0.58	0.90	0.70
(2) RN101	**0.68**	**0.92**	**0.80**	**0.64**	**0.92**	**0.79**
(3) RN101	0.52	0.67	0.61	0.48	0.67	0.58
(4) RN101	0.46	0.58	0.53	0.42	0.58	0.51
(5) RN50	0.45	0.56	0.53	0.42	0.56	0.51
(6) RN50	0.42	0.53	0.49	0.39	0.52	0.48
(7) RN50	0.41	0.50	0.48	0.38	0.51	0.47
(8) RN50	0.40	0.49	0.47	0.37	0.49	0.45
(9) RN152	0.39	0.49	0.46	0.37	0.48	0.45
(10) RN152	0.39	0.48	0.46	0.36	0.48	0.44
(11) RN152	0.39	0.48	0.45	0.36	0.48	0.44
(12) RN152	0.39	0.48	0.45	0.36	0.48	0.45

TABLE IV: RESULTS ON THE HIGH-RESOLUTION TEST SET

Experiment	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{mask}	AP^{mask}_{50}	AP^{mask}_{75}
(1) RN101	0.57	0.86	0.71	0.55	0.85	0.68
(2) RN101	**0.59**	**0.87**	**0.76**	**0.57**	**0.87**	**0.70**
(3) RN101	0.50	0.71	0.64	0.46	0.71	0.57
(4) RN101	0.43	0.59	0.54	0.39	0.59	0.49
(5) RN50	0.40	0.56	0.50	0.37	0.55	0.45
(6) RN50	0.35	0.52	0.43	0.34	0.52	0.42
(7) RN50	0.37	0.54	0.45	0.36	0.55	0.45
(8) RN50	0.33	0.47	0.39	0.32	0.48	0.39
(9) RN152	0.56	0.86	0.67	0.52	0.86	0.65
(10) RN152	0.58	0.85	0.73	0.52	0.85	0.64
(11) RN152	0.48	0.66	0.58	0.43	0.65	0.55
(12) RN152	0.45	0.61	0.56	0.40	0.60	0.52

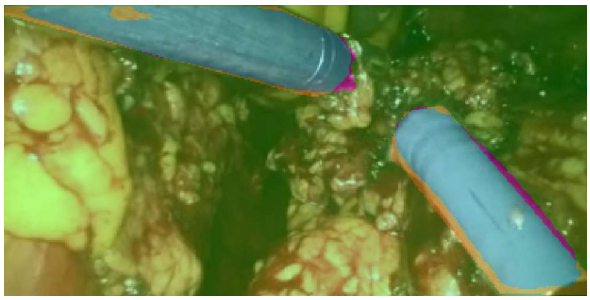
TABLE V: RESULTS ON THE LOW-RESOLUTION TEST SET

Experiment	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{mask}	AP^{mask}_{50}	AP^{mask}_{75}
(1) RN101	0.18	0.40	0.10	0.09	0.28	0.04
(2) RN101	**0.26**	**0.49**	0.25	**0.24**	**0.47**	**0.22**
(3) RN101	0.24	0.42	**0.26**	0.21	0.39	0.21
(4) RN101	0.22	0.39	0.24	0.20	0.37	0.20
(5) RN50	0.22	0.39	0.25	0.20	0.37	0.20
(6) RN50	0.22	0.39	0.24	0.20	0.37	0.20
(7) RN50	0.22	0.38	0.23	0.20	0.37	0.20
(8) RN50	0.21	0.38	0.23	0.20	0.37	0.21
(9) RN152	0.21	0.38	0.22	0.20	0.37	0.20
(10) RN152	0.21	0.37	0.22	0.20	0.36	0.20
(11) RN152	0.21	0.36	0.24	0.19	0.35	0.19
(12) RN152	0.21	0.35	0.24	0.19	0.34	0.19

when the IOU overlap threshold is set to 0.75, 80% and 76% on boxes on validation and high-res test sets respectively, and 79% and 70% on masks.

Despite presenting visual artefacts, the results on the high-res test set show that Mask R-CNN is able to generalize well and excludes those artefacts from the segmentation. At the same time, the performance on the low-res test set degraded for all the models, with the highest AP^{bb}_{50} being 49%, and the highest AP^{mask}_{50} reaching 47%. While those scores are worse on the other datasets, they are still acceptable and show once again that the network trained in experiment 2 did not overfit the training set.

In general, we found that the quality of the prediction varies depending on the scene, quality of the image, position of the tools (e.g. tool occlusions), getting the worst results in situations that were not present in the training set, such as crossing tools or when an organ or tissue in the image has a similar color and shape as a tool. In figure 3, examples of good and bad predictions are shown. The most common errors include predicting wrong elements as tools, and the prediction of overlapped tools.



(a)

(b)

Fig. 4: Mask R-CNN prediction examples vs ground truth. True Positives: cyan, False Positives: magenta, False Negatives: yellow, True Negatives: green. Image 4a shows the output from the model trained in experiment number 2 (best performance), while image 4b shows the output from model number 11 (worst performance on validation set).

In Figure 4 it is possible to observe the slight difference between the mask predictions and the ground truth on the same image for two different models. True/false positives and true/false negatives are highlighted in the image.

Regarding data augmentation, by comparing the results of experiments 1-2, 5-6 and 9-10 we can see that the ResNet101-

based model shows an improvement on all datasets using a training dataset with augmented images, highlighting the importance of this technique. The other models did not benefit from data augmentation. By comparing experiments 3-4, 7-8 and 11-12, we can also see that all the models seem to prefer a weaker L2 regularization parameter of 0.0001.

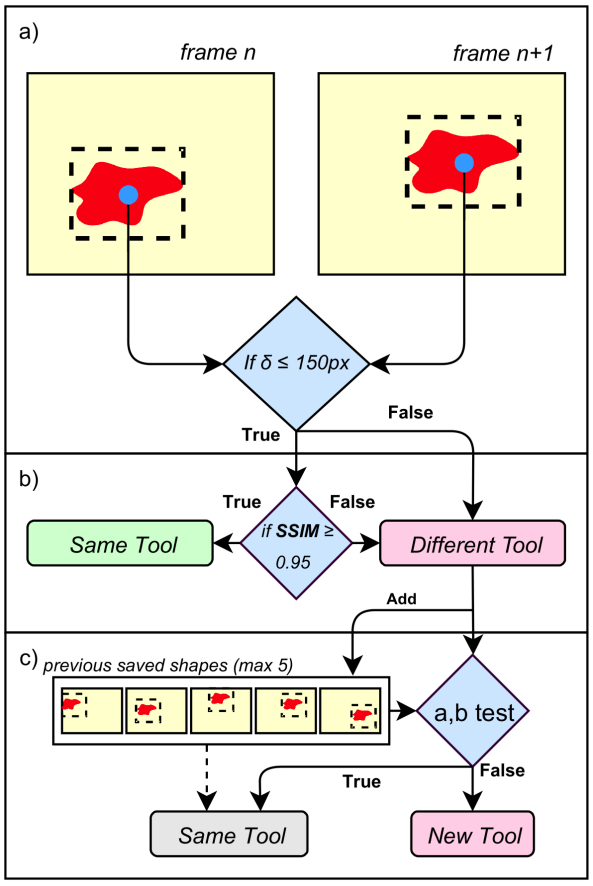


Fig. 5: The pipeline to track the tools in consecutive frames is shown here. The process is performed by computing the similarity among the binary masks of the tools and with between-frames positional center similarities. In a) the Euclidean distances between the bounding box centers are computed. Then in b) a test with the *Structural similarity index* (**SSIM**) between the masks of the tools is computed. If the test is not passed, the tool is then compared to the last 5 saved tools (c), like in a) and b). The tools are considered the same and are visualized with the same colors if and only if the Euclidean distance between the centers is below a certain threshold (150 px in our case) and the SSIM is greater than or equal to a second threshold (0.95 in our example). When a new tool is found, it is buffered to the saved pool of tools.

D. Segmentation qualitative analysis as a base for tracking methods

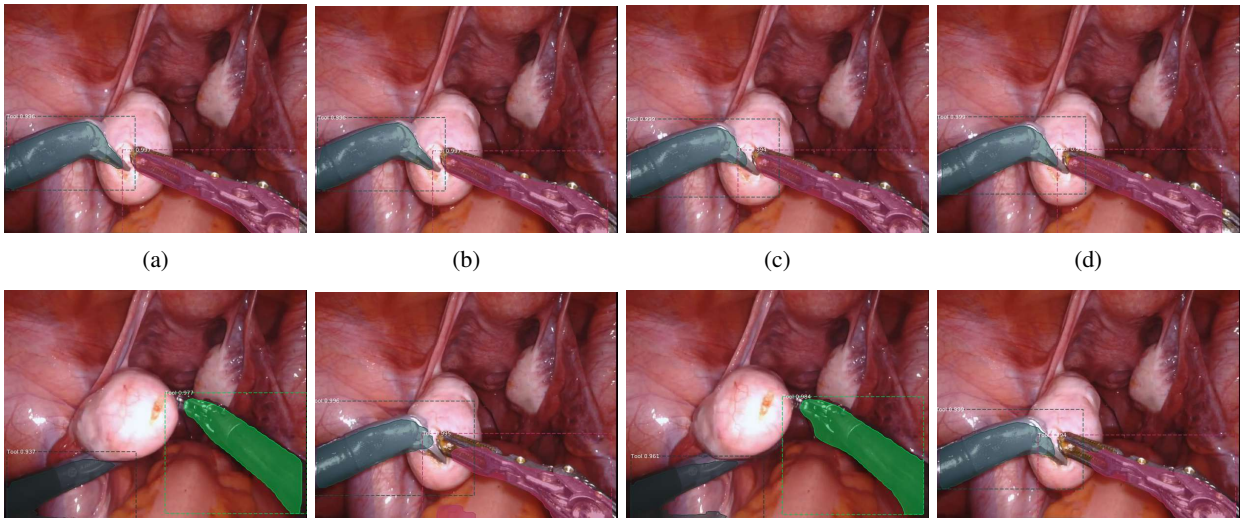
In order to evaluate the possibility of employing the segmentation output as a basis for surgical tool tracking, a basic tracking algorithm is implemented. Since annotations for the tracking task are not currently available, a qualitative analysis is performed by experts on the output of the tracking algorithm. The implemented tracking procedure is based on a two-phase comparison: i) evaluation of the proximity of the bounding boxes, ii) the computation of the structural similarity index (SSIM) [42] between the two frame-adjacent masks. The between-frames bounding box proximity is computed between adjacent frames and it is a necessary condition for applying segmentation-based tool similarity comparisons.

In particular, for each tool in a new frame, a comparison is made with the tools detected in the previous frame. If the Euclidean distance between the centers of the bounding boxes is less than a certain threshold (empirically fixed to 150 pixels in the high-resolution image case) on both axes, the selected boxes' masks are compared with SSIM against a specific threshold. In our case, the bounding boxes are considered to outline the same segmented tool only if $SSIM(m_1, m_2) \geq 0.95$, with m_1 and m_2 being the two compared masks. If the tool is not found in the previous frame, it is compared to a pool of previously encountered tools, in order to recover the identity of tools that have disappeared in the previous frames (or that have not been detected by the Mask R-CNN). The two previously-described comparisons are then repeated using the previously saved bounding boxes and masks to try to find a match. If a match is not found, the tool is considered a new tool and is added to the pool of saved tools. We decided to limit the size of the pool to 5. Both the Euclidean distance threshold and the SSIM threshold were chosen empirically. The choice of using the SSIM was inspired by recent works that employed it in other segmentation tasks [43] and by its use in some tracking algorithms [44]. The SSIM was computed using the function *compare_ssim*, available in the *skimage* Python package.

In short, the tracking algorithm recognizes the shape of similar segmented tools that are close in space in adjacent frames and assigns the same identity to such tools. In Figure 6 a sequence of surgical tools is segmented and tracked by applying this methodology. A different color is used to distinguish the different tools. In this specific scene, four different tools are show up at a different time in the sequence. The identities of the masks have been observed along the whole sequences by three experts and the tracking output was judged to be of good quality. A quantitative tracking evaluation is planned to be performed in the future.

V. Conclusion

We have proposed the use of Mask R-CNN to detect and segment surgical tools into endoscopic/laparoscopic images. We trained and evaluated Mask R-CNN with different backbone structures (ResNet50, ResNet101 and ResNet152) along with different data augmentation techniques and hyperparameter tuning. After evaluation, the proposed approach shows good potential for use in endoscopic surgical tools detection and segmentation, as well as being a solid base for the implementation of a tracking algorithm. The best results on our dataset were obtained training the network with a ResNet101



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 6: Example of results of the tracking algorithm, each tool is recognized and segmented using a specific color. From figure 6a to figure 6d two objects are recognized (in gray and purple), the same objects are correctly recognized in frame 6f and 6h. In figure 6e and 6g. two equal tools are alternated with the previous tools and different colors (black and green) are assigned.

backbone for 25 epochs. Moreover, we showed that the trained model was robust to image artefacts and could still work reasonably well on low-resolution images. However, the model still presented some limitations and failure cases, opening the way to possible future improvements, such as the use of a bigger and richer training dataset. A quantitative tracking evaluation, along with more complex tracking algorithms, should also be explored in future research.

REFERENCES

[1] N. Mammone, C. Ieracitano, and F. C. Morabito, "A deep cnn approach to decode motor preparation of upper limbs from time–frequency maps of eeg signals at source level," *Neural Networks*, vol. 124, pp. 357–372, 2020.

[2] C. Ieracitano, N. Mammone, A. Bramanti, A. Hussain, and F. C. Morabito, "A convolutional neural network approach for classification of dementia stages based on 2d-spectral representation of eeg recordings," *Neurocomputing*, vol. 323, pp. 96–107, 2019.

[3] M. Zhou, C. Tian, R. Cao, B. Wang, Y. Niu, T. Hu, H. Guo, and J. Xiang, "Epileptic seizure detection based on eeg signals and cnn," *Frontiers in neuroinformatics*, vol. 12, p. 95, 2018.

[4] 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.

[5] J. Kang and J. Gwak, "Ensemble of instance segmentation models for polyp segmentation in colonoscopy images," *IEEE Access*, vol. 7, pp. 26 440–26 447, 2019.

[6] X. Mo, K. Tao, Q. Wang, and G. Wang, "An efficient approach for polyps detection in endoscopic videos based on faster r-cnn," in *2018 24th International Conference on Pattern Recognition (ICPR)*. IEEE, 2018, pp. 3929–3934.

[7] Y. Shin, H. A. Qadir, L. Aabakken, J. Bergsland, and I. Balasingham, "Automatic colon polyp detection using region based deep cnn and post learning approaches," *IEEE Access*, vol. 6, pp. 40 950–40 962, 2018.

[8] I. J. Goodfellow, Y. Bulatov, J. Ibarz, S. Arnoud, and V. Shet, "Multi-digit number recognition from street view imagery using deep convolutional neural networks," *arXiv preprint arXiv:1312.6082*, 2013.

[9] K. Rohit Malhotra, A. Davoudi, S. Siegel, A. Bihorac, and P. Rashidi, "Autonomous detection of disruptions in the intensive care unit using deep mask r-cnn," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 1863–1865.

[10] S. Ali, F. Zhou, C. Daul, B. Braden, A. Bailey, S. Realdon, J. East, G. Wagnières, V. Loschenov, E. Grisan *et al.*, "Endoscopy artifact detection (ead 2019) challenge dataset," *arXiv preprint arXiv:1905.03209*, 2019.

[11] S. Ali, F. Zhou, A. Bailey, B. Braden, J. East, X. Lu, and J. Rittscher, "A deep learning framework for quality assessment and restoration in video endoscopy," *arXiv preprint arXiv:1904.07073*, 2019.

[12] J. Hung and A. Carpenter, "Applying faster r-cnn for object detection on malaria images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 56–61.

[13] S. Bernhardt, S. A. Nicolau, L. Soler, and C. Doignon, "The status of augmented reality in laparoscopic surgery as of 2016," *Medical image analysis*, vol. 37, pp. 66–90, 2017.

[14] J. Kowalczuk, A. Meyer, J. Carlson, E. T. Psota, S. Buettner, L. C. Pérez, S. M. Farritor, and D. Oleynikov, "Real-time three-dimensional soft tissue reconstruction for laparoscopic surgery," *Surgical endoscopy*, vol. 26, no. 12, pp. 3413–3417, 2012.

[15] M.-C. Dy, K. Tagawa, H. T. Tanaka, and M. Komori, "Method in collision detection and interaction between rigid surgical tools and deformable organs," in *SIGGRAPH Asia 2014 Posters*, 2014, pp. 1–1.

[16] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[17] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[18] Q. You, J. Luo, H. Jin, and J. Yang, "Building a large scale dataset for image emotion recognition: The fine print and the benchmark," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[19] N. Murray, L. Marchesotti, and F. Perronnin, "Ava: A large-scale database for aesthetic visual analysis," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 2408–2415.

[20] D. Acuna, H. Ling, A. Kar, and S. Fidler, "Efficient interactive annotation of segmentation datasets with polygon-rnn++," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 859–868.

[21] A. Dutta and A. Zisserman, "The via annotation software for images,

audio and video," in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 2276–2279.

[22] C. Zhang, K. Loken, Z. Chen, Z. Xiao, and G. Kunkel, "Mask editor: an image annotation tool for image segmentation tasks," *arXiv preprint arXiv:1809.06461*, 2018.

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[24] M. Seyedhosseini, M. Sajjadi, and T. Tasdizen, "Image segmentation with cascaded hierarchical models and logistic disjunctive normal networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 2168–2175.

[25] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[26] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732.

[27] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei, "Fully convolutional instance-aware semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2359–2367.

[28] 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*, 2015, pp. 91–99.

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

[30] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," in *Artificial Neural Networks and Machine Learning – ICANN 2018*, V. Kůrková, Y. Manolopoulos, B. Hammer, L. Iliadis, and I. Maglogiannis, Eds. Cham: Springer International Publishing, 2018, pp. 270–279.

[31] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[32] K. Yu, W. Xu, and Y. Gong, "Deep learning with kernel regularization for visual recognition," in *Advances in Neural Information Processing Systems*, 2009, pp. 1889–1896.

[33] R. Rawat, J. K. Patel, and M. T. Manry, "Minimizing validation error with respect to network size and number of training epochs," in *The 2013 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2013, pp. 1–7.

[34] D. A. Van Dyk and X.-L. Meng, "The art of data augmentation," *Journal of Computational and Graphical Statistics*, vol. 10, no. 1, pp. 1–50, 2001.

[35] L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning," *arXiv preprint arXiv:1712.04621*, 2017.

[36] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[37] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[38] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[39] A. B. Jung, "imgaug," https://github.com/aleju/imgaug, 2018, [Online; accessed 30-Oct-2018].

[40] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*. USA: McGraw-Hill, Inc., 1986.

[41] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Simultaneous detection and segmentation," in *European Conference on Computer Vision*. Springer, 2014, pp. 297–312.

[42] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, vol. 2. Ieee, 2003, pp. 1398–1402.

[43] S. Zhao, B. Wu, W. Chu, Y. Hu, and D. Cai, "Correlation maximized structural similarity loss for semantic segmentation," *arXiv preprint arXiv:1910.08711*, 2019.

[44] A. Loza, L. Mihaylova, N. Canagarajah, and D. Bull, "Structural similarity-based object tracking in video sequences," in *2006 9th International Conference on Information Fusion*. IEEE, 2006, pp. 1–6.