Journal of Medical Robotics Research, Vol. 4, No. 3 (2019) 1942005 (9 pages)

DOI: 10.1142/S2424905X19420054

Journal of Medical Robotics Research
https://www.worldscientific.com/worldscinet/jmrr

Detection of Suture Needle Using Deep Learning

Qipei Mei*, Jonathan Chainey†, David Asgar-Deen‡, Daniel Aalto§,¶

*Civil and Environmental Engineering
University of Alberta, 7-203 Donadeo Innovation Centre for Engineering
9211 116 Street NW, Edmonton, Alberta, Canada, T6G 1H9

†Medicine and Dentistry, University of Alberta
2J2.00 Walter C Mackenzie Health Science Centre, 8440 112 Street NW
Edmonton, Alberta, Canada, T6G 2R7

‡Electrical and Computer Engineering, University of Alberta
7-203 Donadeo Innovation Centre for Engineering
9211 116 Street NW, Edmonton, Alberta, Canada, T6G 1H9

§Communication Sciences and Disorders, Rehabilitation Medicine
University of Alberta, Clinical Sciences Building
11304 83 Avenue NW, Edmonton, Alberta, Canada, T6G 2G3

¶Institute for Reconstructive Sciences in Medicine, Misericordia Community Hospital
1W-02, 16940-87 Ave, Edmonton, Alberta, Canada, T5R 4H5

The importance of surgical simulation has increased over the last decade and the majority of medical schools have incorporated simulation into their curriculum. An essential aspect of surgical education is to evaluate how the student performs when compared to an expert surgeon. Another way to evaluate the skill of the student would be by tracking the position of the needle during the procedure, a factor correlating to surgical skill. In this study, we developed deep learning algorithms for needle detection during a video of a surgical procedure. 78 videos of a person doing a running suture on synthetic skin were captured using an HD camera. A total of 3368 images were manually annotated with a VGG annotator tool. Two deep learning algorithms (YOLOv3 and Faster R-CNN) were pretrained on 2219 images extracted from the JIGSAWS dataset, then trained on the 804 images from the training set and finally applied to the 345 images from the evaluation set. The performance of the algorithm was evaluated using the intersection over union (IoU) method as well as by measuring the Euclidean distance between bounding box centroids. These values were compared against the inter-observer reliability among three authors. The best IoU value by deep learning algorithms compared against the ground truth was found to be 0.601 for Faster R-CNN while the average inter-observer value was 0.663. The average Euclidean distances between bounding box centroids for authors and for the Faster R-CNN algorithm were 21.9 pixels and 36.8 pixels, respectively. Through qualitative and quantitative assessment of the algorithm (visually observing the algorithm's needle annotations), deep learning shows promise for automatically tracking the position of the needle during a suturing operation.

Keywords: Deep learning; object detection; needle tracking; suture; image processing.

JMRR

1. Introduction

One of the first surgical procedures a medical student learns to perform is suturing. This is often done in a simulation environment to ensure a safe and effective learning environment. Thus, simulation is becoming an important part of health care education. The current practice in medical school is to have a supervisor (a board-certified

Received 30 July 2019; Revised xx xx xx; Accepted 6 January 2020; Published 17 April 2020. Published in JMRR Special Issue on Technology-enabled Tools for Human Skill Assessment. Guest Editor: Joseph Singapogu.
Email Address: ¶*aalto@ualberta.ca*

surgeon or a surgical resident) present during the surgical simulation to correct and modify the learner's movements. This can be expensive and time limiting.

Comparing the surgical performance of a medical student against an expert surgeon's performance is a valuable teaching tool as it allows one to find areas in which the medical student can improve [1]. In order to compare performances, we first need to establish objective metrics that constitute a surgical performance. Criteria-based assessment tools such as the Objective Structured Assessment of Technical Skills (OSATS) have been developed to evaluate surgical performance [2,3]. This has been further improved by removing the subjectivity of human evaluation by incorporating automated surgical skills assessments [4,5].

Another area where the computer sciences have been used to characterize a surgeon's skills is instrument tracking. One method currently used for instrument tracking is based on infrared optical markers [6–9]. However, this technology has some limitations. It is not suited to detect the position of the needle as the markers would need to be located on the instruments manipulating the needle offering indirect information about the tissue-needle interaction. As knowledge about video analysis in surgical education has been growing, different methods have emerged [9–11]. For example, Kranzfelder *et al.* [12] used radiofrequency identification technology to detect instrument position in minimally invasive surgery. Image-based analysis applying segment and contour processing as well as 3D modeling have also been reported. Other studies have used deep learning approaches based on convolutional neural networks (CNN) for surgical tools detection [13–17].

The scientific literature has mainly focused on the position and motion of the surgical instruments, neglecting to monitor the position of the needle during the surgical task. This metric is a useful marker of a surgeon's skills. Improper needle position can result in increased regrasps with the needle driver, task completion times and unsuccessful attempts [18] and is related with more tissue trauma [19].

Deep learning-based object detection and tracking algorithms have attracted much attention in recent years due to their superior performance when compared to more traditional methods. In 2014, Girshick *et al.* [20] proposed a method called 'Region-based Convolutional Network' (R-CNN). In this method, they introduced the concept of region proposals and two-step detection. They first generated a series of candidate bounding boxes and then did classification and regression on these bounding boxes. Inspired by R-CNN, a number of improved algorithms were suggested. For instance, fast R-CNN and faster R-CNN [21,22] improved the classification and bounding box regression tasks and achieved a better performance and lower computational time. Single shot multibox detect (SSD) [23] discretized the output space into bounding boxes with different scales and aspect ratios, and conducted object detection using features from different levels of the neural network. 'You Only Look Once' (YOLO) is another popular method inspired by R-CNN which can do the object detection in real time with acceptable accuracy. YOLO was first proposed by Redmon *et al.* [24] in 2016. With generations of improvements, Faster R-CNN and YOLOv3 are now among the fastest and most accurate object detection algorithms.

In this paper, Sec. 2 will talk about deep learning based methodologies, Sec. 3 will present the experiment and results. Discussion and conclusions are included in Secs. 4 and 5. The contribution of this paper is the introduction of two deep learning algorithms, Faster R-CNN and YOLOv3, for suture needle detection in terms of surgical education, and the comparison of their performance with human annotators.

2. Methodology

2.1. *Faster R-CNN*

Faster R-CNN is the latest version of the R-CNN family [20,22,25]. Instead of using selective search to generate proposed regions, Faster R-CNN utilizes CNNs for both region proposal and object detection. This configuration can significantly increase the computational efficiency. As presented in Fig. 1, there are four components in Faster R-CNN: a feature extraction network, a region proposal network (RPN), region of interest (ROI) pooling layers and detection layers.

The feature extraction network in this study is a pretrained ResNet50. The ResNet50 consists of convolutional layers and skip connections. The convolutional layer is a commonly used layer in deep learning and was first proposed by LeCun *et al.* [26]. Unlike fully connected layers in traditional neural networks, a convolutional layer uses a sliding window to scan through the image to do the convolution operations. The sliding window is similar to a regular filter except the weights of it are determined by the training of the neural network. Owing to the characteristics of the convolutional layer, it has been successfully applied to a variety of tasks in computer vision [24,27]. The skip connections that connect two nonconsecutive layers are introduced by He *et al.* to resolve the training issue [28]. The layer within a skip connection is called a residual block. These residual blocks can help the training of the deep neural network and can lead to better performance.

The features maps extracted from ResNet50 are then fed into the region proposal network. A sliding window is applied in the RPN to go through each location over the feature maps. Anchor boxes are generated for these locations. There are two branches in the RPN where one predicts whether the box has objects or not and the other makes a regression to provide a better estimate of the box

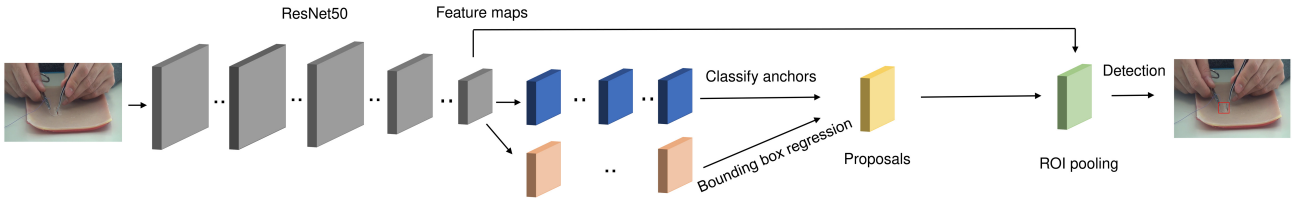


Fig. 1. Architecture of faster R-CNN.

coordinates. The loss function of the RPN is described in Eq. (1).

$$L_{\mathrm{RPN}}(p_i, t_i) = \frac{1}{N_{\mathrm{cls}}} \sum_i L_{\mathrm{cls}}(p_i, p_i^*) + \lambda \frac{1}{N_{\mathrm{reg}}} \sum_i p_i^* L_{\mathrm{reg}}(t_i, t_i^*), \quad (1)$$

where p_i and p_i^* are the predicted and ground truth labels for the anchor i, and t_i and t_i^* are the predicted and ground truth coordinates for the anchor i. L_{cls} is the log loss and L_{reg} is the regression loss defined in [25].

These two branches in RPN are merged after non-maximum suppression to propose a reasonable number of potential bounding boxes. After that, ROI pooling layers are applied to collect proposal boxes and original feature maps to standardize the sizes of the proposal boxes. At last, two fully connected branches are applied for classification and bounding box regression. More details of Faster R-CNN can be found in [22].

2.2. *YOLOv3*

YOLOv3 [30], the latest version of YOLO, is among the most accurate and fastest object detection/tracking algorithms [31]. As a deep learning-based method, YOLOv3 is efficient at detecting objects at varying conditions without handcrafted features. YOLOv3 consists of convolutional and skip connections. This configuration increases the computational efficiency and the ability to detect small objects using multi-level prediction. The architecture of the YOLOv3 algorithm for needle tracking is presented in Fig. 2. In total, YOLOv3 has 106 layers, most of which are convolutional layers.

The backbone model of YOLOv3 is a 74-layer deep neural network (including skip connections) called Darknet53 proposed by Redmon and Farhadi [30]. The reason it is called Darknet 53 is because the number of layers excluding skip connections is 53. An additional 32 layers are wired to original Darknet53 to accomplish the prediction function. It can be seen that this neural network consists of 23 residual blocks where each residual block has a skip connection. The details of skip connections will be presented in an upcoming section. In YOLOv3, Darknet53 is first trained on a large scale dataset ImageNet [32] and then used as backbone model to support further detection of the suture needle.

The convolutional layers are applied to the image by scanning all the possible regions. At the end of each convolutional layer, the Leaky Rectified Linear Units (ReLU) layer follows to increase the nonlinearity of the model. Among all the layers, YOLO layers at 82, 94 and 106 are in charge of bounding box prediction. The cell

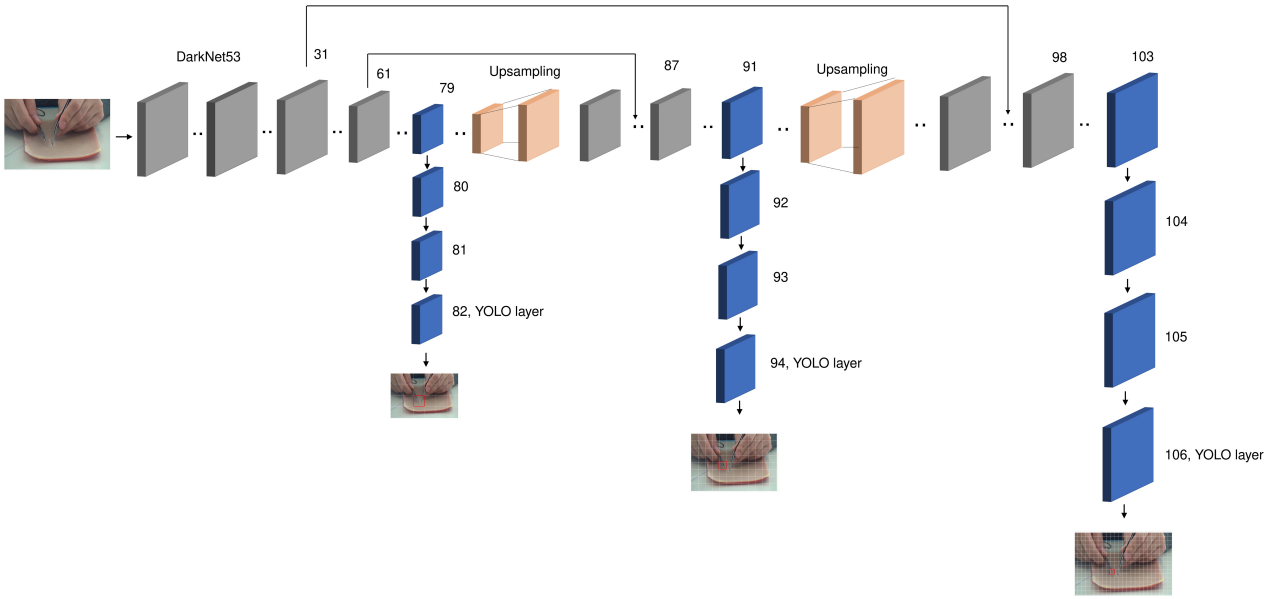


Fig. 2. Architecture of YOLOv3 (adapted from [29]).

t_x	t_y	t_w	t_h	p_0	p_1

Fig. 3. Format of a bounding box prediction.

where the center of the needle is located is in charge of the bounding box prediction of that needle (red bounding box in Fig. 2). The resolutions of predictions in these three layers are different, where layer 82 is the roughest and 106 is the finest. This design can help predict the object with varying sizes. The algorithm will combine all three features maps together to give a final prediction.

As described above, the YOLO layers are in charge of prediction. The output of a YOLO layer is generated by applying a $1 \times 1 \times 18$ kernel to the previous layer. Each cell can predict 3 bounding boxes, so each bounding box has six parameters. The format of each bounding box is presented in Fig. 3 where t_x, t_y, t_w, t_h are the parameters for the x, y coordinates and the width and height of the bounding box, and p_0, p_1 are the confidence and class score for this prediction. To convert the bounding box parameters to real coordinates and dimensions, the following Equation is used [30].

$$\begin{aligned} b_x &= \sigma(t_x) + c_x, \\ b_y &= \sigma(t_y) + c_y, \\ b_w &= p_w e^{t_w}, \\ b_h &= p_h e^{t_h}, \end{aligned} \tag{2}$$

where $\sigma(t)$ is the sigmoid function, c_x and c_y are the indices of the cell, p_w and p_h are the scaled anchors, b_x and b_y are the real coordinates of the bounding box, and b_w and b_h are the real width and height of the bounding box.

The loss function is the function that is being minimized during the training process. In YOLOv3 used in this paper, there are three kinds of loss: coordinate loss, dimension loss and confidence loss. Since there is one class in our object detection task, there is no loss coming from class assignment. The total loss is the summation of loss from all three YOLO layers.

$$\begin{aligned} L_{\text{coordinate},k} &= \sum_{i=0}^{S_k^2} \sum_{j=0}^{B} \delta_{ij} \left[(x_{ij} - x_{ij}^*)^2 + (y_{ij} - y_{ij}^*)^2 \right], \\ L_{\text{dimension},k} &= \sum_{i=0}^{S_k^2} \sum_{j=0}^{B} \delta_{ij} \left[\left(\sqrt{w_{ij}} - \sqrt{w_{ij}^*} \right)^2 \right. \\ &\quad \left. + (\sqrt{h_{ij}} - \sqrt{h_{ij}^*})^2 \right], \\ L_{\text{confidence},k} &= \sum_{i=0}^{S_k^2} \sum_{j=0}^{B} \delta_{ij} \left[(C_{ij} - C_{ij}^*)^2 \right] \\ &\quad + \lambda_{\text{no-obj}} \sum_{i=0}^{S_k^2} \sum_{j=0}^{B} (1 - \delta_{ij}) \left[(C_{ij} - C_{ij}^*)^2 \right], \\ L_{\text{total}} &= \sum_{k=1}^{3} (L_{\text{coordinate},k} + L_{\text{dimension},k} + L_{\text{confidence},k}). \end{aligned} \tag{3}$$

In Eq. (3), k stands for the kth YOLO layer, S_k is the number of cells in one dimension for the scale k, and B is the maximum bound boxes the algorithm can predict which is 3 in our case. In addition, δ_{ij} is 1 when there is an object in cell i for bounding box j and 0 otherwise. $\lambda_{\text{no-obj}}$ is the weight given to the confidence loss for cells that do not have an object. The variables x_{ij}, y_{ij}, w_{ij}, h_{ij}, C_{ij} are the predicted values of the x, y coordinates, width, height and confidence score of the object, respectively. The variables with an asterisk denote ground truth.

3. Experiment

3.1. *Data preparation*

One of the authors performed running sutures of a single incision measuring 10 cm long on a synthetic skin (pocket 3-layer suture pad with clear case $3.75'' \times 2.75''$, Your Design Medical, Brooklyn, NY) using a type 2-0 vicryl suture needle. Tissue bites were taken on either side of the wound at 1 cm intervals resulting in approximately 8 or 9 stitches. For the training set, 78 videos ranging from 9 s to 25 s, were captured with an HD camera (HDR-CX360V, Sony, Tokyo, Japan) stabilized on a stand. During the recording, each video was taken with some variation including: different angles (increment of 45° around the surgical table for each video), different levels of zoom (ranging from no zoom to 2× zoom), and different backgrounds and lighting environments by recording in different rooms. This was done to recreate a realistic surgical environment where the camera is not fixed. In addition, a wide variation in the images of our training set would not limit the medical student to a specific camera setting while practicing surgical sutures. The best 50 videos, defined as videos where the needle was not occluded for the majority of the duration, were kept to be further processed. Using a custom Matlab algorithm, a frame was captured every 0.5 s starting at 0 s for the first 8 s of each video resulting in 850 images. It should be noted that some images contained no information and were subsequently dropped with 804 images remaining for the training set. In addition to the suturing videos made by the authors, suturing videos from the open access JHU-ISI Gesture and Skill Assessment Working Set (JIGSAWS) were used in the training set [33]. Using the same Matlab algorithm, a frame was captured every 0.5 s of the video and 2500 images were extracted from the suturing videos of the JIGSAWS set. Among the 2500 images, only 2219 images include needles, which were then used for training. In total, $804 + 2219 = 3023$ images were used in the training set.

For the evaluation set, a new synthetic skin with a single incision measuring 10 cm was used. One video of 194 s was recorded while performing a running suture

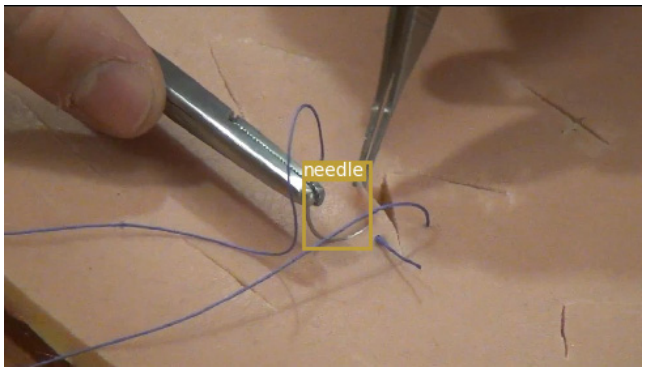


Fig. 4. Sample prediction from deep learning algorithms.

over the incision using the same camera. There was no angle or zoom variation for this take. Using the same Matlab algorithm, 388 images were retrieved from the evaluating video and 345 were kept after discarding frames with no information. A total of 3368 images were manually annotated by the authors using the VGG annotator tool [26]. To perform the annotation, a rectangular box was drawn manually around the needle to identify the location of the needle on each image. This rectangle is called the ground-truth bounding box as presented in Fig. 4. In summary, 3023 images were used for training and 345 images were used for evaluation. It should be noted that only 276 images in the evaluation set had a needle visible in the frame (according to the annotators).

3.2. *Evaluation of the algorithm performance*

To measure the performance of the needle tracking algorithm, intersection over union (IoU) is used to evaluate the similarity of two bounding boxes. The equation used to calculate the IoU can be found in Eq. (4). The numerator, A_{overlap}, is the area of overlap between two bounding boxes while the denominator, A_{union}, is the total area of union created by both bounding boxes. If two bounding boxes are perfectly matched, the IoU is 1. In contrast, if they have no overlap, the IoU is 0. It should be noted that the IoU value is only calculated when both the author and algorithm agreed that a needle was in the frame.

$$I_{\text{IOU}} = \frac{A_{\text{overlap}}}{A_{\text{union}}}. \tag{4}$$

In addition to IoU, the mean Euclidean distance of the bounding box centroids is used to measure performance. Similar to the IoU algorithm, the Euclidean distance could only be calculated if both the annotator and the algorithm detected a needle in a frame.

3.3. *Results and analysis*

The gold standard for performance is set as the IoU and the mean Euclidean distance between bounding box centroids between the authors (QM, JC, DAD), in other words the inter-observer reliability. These values were gathered by comparing the annotations of 276 unique images. As each image has its own IoU value, the IoU value that will be referred to in this paper will relate to the average value over the set of images. It should also be noted that if there was no needle in the frame (no bounding boxes), no IoU value would be assigned. This means if any two users agreed there was no needle (true negative), that specific observation would not affect the mean IoU value. The IoU values between the three possible annotator pairs were 0.620, 0.663 and 0.691. The mean Euclidean distance between bounding box centroids between annotators was 19.6, 21.9, and 25.4 pixels, respectively. It should be noted that when annotators detected a needle, there was always some overlap between the bounding boxes.

The deep learning algorithms are trained on a desktop PC with i7-8700 CPU, 32 GB memory and Nvidia Titan V GPU. The learning rate of 0.001 is used. The input images are resized to 416×416 pixels. The batch size is set to 12 considering the memory limitation of GPU. The adam optimizer with a momentum of 0.9 is used.

Figures 5 and 6 show the mean IoU given by each algorithm for different epochs. The solid line shows the gold standard, i.e. the average IoU value between all three authors (0.663). Figures 7 and 8 illustrate the percentage of images that were annotated correctly (true positive) over the total amount of images with a needle apparent (276 images). A 100% proper classification would indicate that the algorithm defined a bounding box every time the author did.

As shown in Fig. 5, the largest mean value (relating to the most overlap in the bounding boxes) was found with

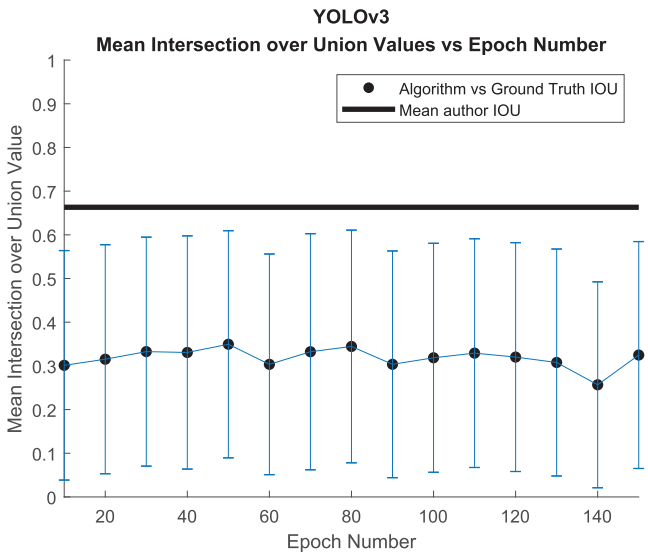


Fig. 5. Mean IoU value given the algorithm's epoch number for the YOLOv3 algorithm. The highest IoU value was 0.349 ± 0.260 at an Epoch value of 50.

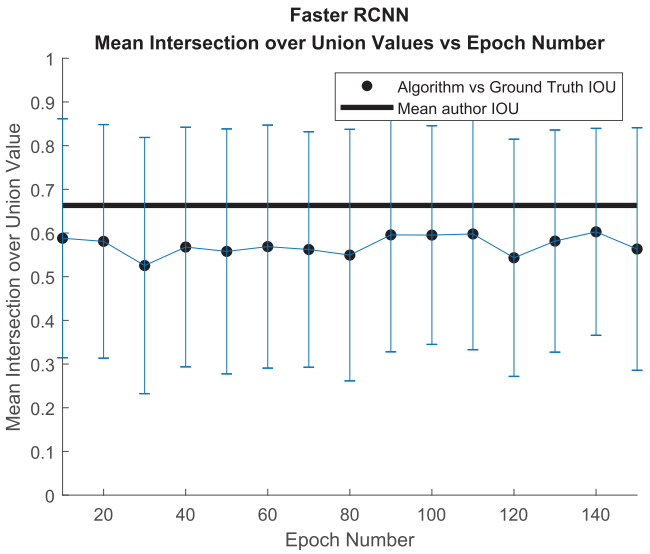


Fig. 6. Mean IoU value given the algorithm's epoch number for the Faster RCNN algorithm. The highest IoU value was 0.6 01 ± 0.237 at an Epoch value of 140.

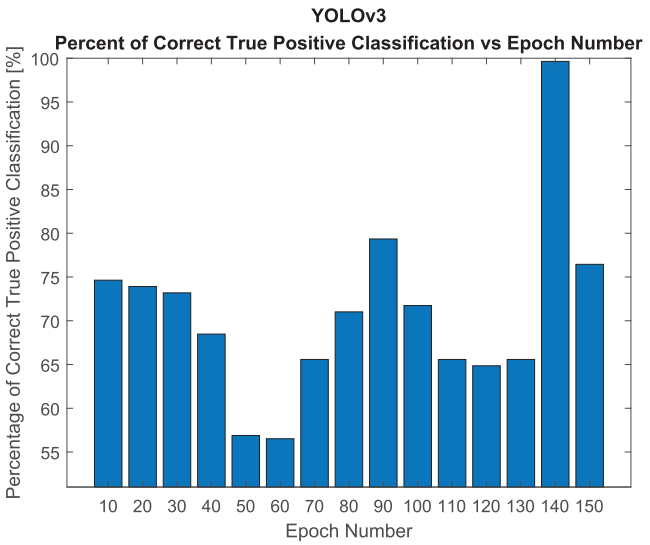


Fig. 7. Percentage of correct true positive classifications given each epoch value for YOLOv3.

an epoch value of 50 with a mean IoU value of 0.349 ± 0.260 for the YOLOv3 algorithm. The number of images used to determine the IoU values between the ground truth and the algorithm for this epoch was 157 images (the number of true positive results). This results in a 56.88% proper classification rate as seen in Fig. 7.

Similarly, Fig. 6 shows the largest mean value of the IoU (0.601 ± 0.237) at an epoch value of 140. This relates to a proper classification rate of 50.7% (140 images used).

Figures 9 and 10 demonstrate the two-norm (Euclidean) distance of predicted and ground-truth bounding boxes for different epochs. The blue points represent the mean value for all the true positive results,

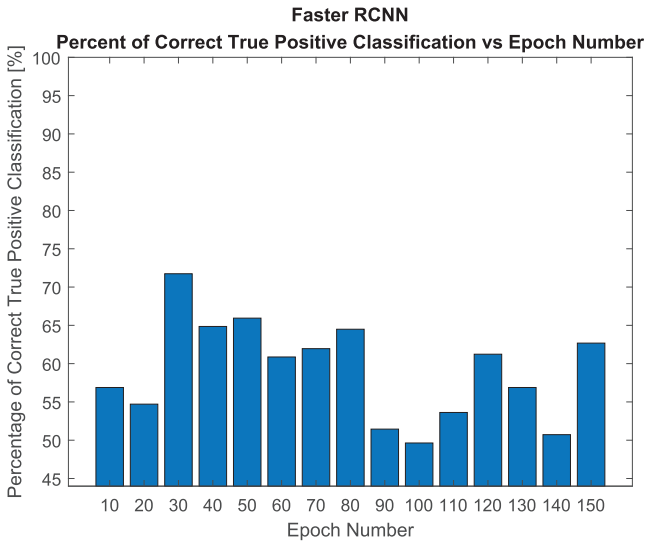


Fig. 8. Percentage of correct true positive classifications given each epoch value for Faster R-CNN.

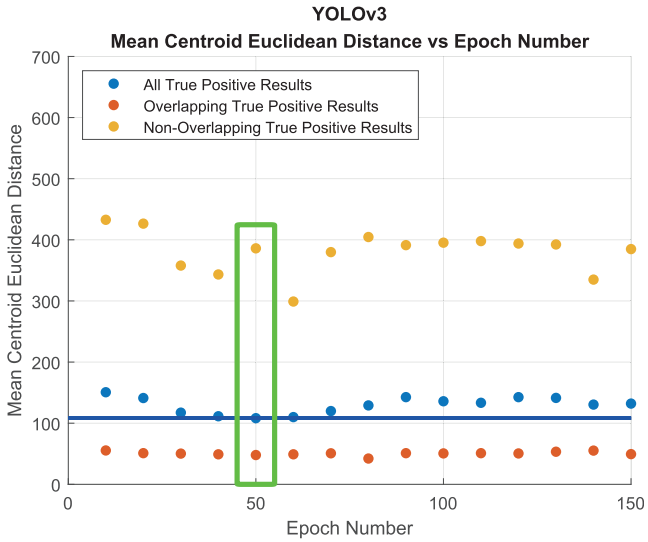


Fig. 9. Mean two-norm distances of the bounding boxes given its Epoch number for YOLOv3. The green box represents the version of the algorithm with the best (lowest) mean distance in pixels. The minimum overall distance of 108.230 pixels can be found at an Epoch value of 50.

the red points represent the mean value for all the overlapping true positive results, and the yellow points represent the mean value for all the nonoverlapping true positive results.

The data collected from the Euclidean distance can be dissected into two sub-categories: images with overlapping bounding boxes and images with no overlapping bounding boxes. By splitting the true positive results into these two categories, insight can be gained regarding how far away the bounding boxes are when there is no overlap, and how close the bounding boxes are when there is overlap.

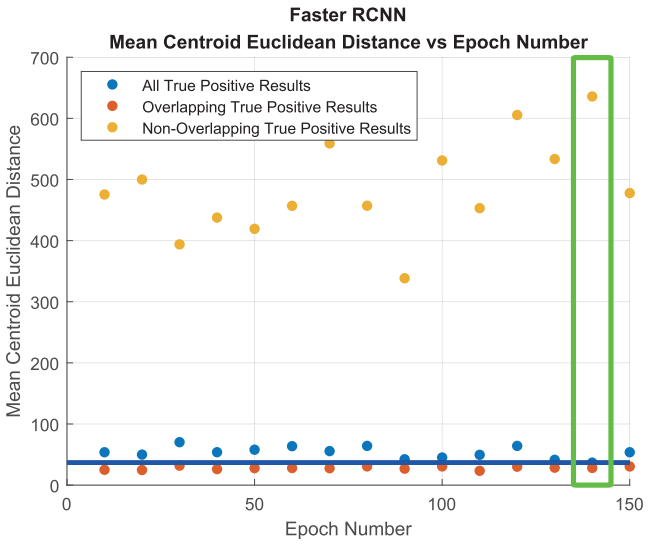


Fig. 10. Mean two-norm distances of the bounding boxes given its Epoch number for Faster RCNN. The green box represents the version of the algorithm with the best (lowest) mean distance in pixels. The minimum overall distance of 36.822 pixels can be found at an Epoch value of 140.

As shown in Fig. 9, the lowest mean Euclidean distance for the YOLOv3 algorithm was found at an epoch of 50, represented by the values in the green box. The total number of true positive results between the ground truth and the algorithm was 157, where 129 of those bounding boxes overlapped and 28 boxes did not overlap. The mean distance between the bounding box centroids was 108.230 pixels, with distances of 47.927 and 386.057 for overlapping and nonoverlapping results, respectively.

Similarly, Fig. 10 shows the lowest mean Euclidean distance for the Faster RCNN algorithm. The lowest mean distance was 36.822 pixels at an epoch value of 140. It should be noted that the mean Euclidean distance stayed relatively constant throughout the epochs (standard deviation of 9.579 for the 15 unique epochs). The distances for the overlapping and nonoverlapping bounding boxes were 28.141 and 635.814, respectively.

Lastly, the average time it took authors QM, JC, and DAD to annotate each image was 5.74, 6.58 and 5.25 s, respectively. The average time it took the Faster RCNN algorithm to process each image was 0.05 s and it took 0.02 s for the YOLOv3 algorithm to process each image.

4. Discussion

In this paper, Faster R-CNN and YOLOv3 algorithms were implemented and shown to successfully identify and localize the needle in video footage of suturing. The performance was measured by comparing it with that of human annotators.

A bounding box was used for annotating the needle location. A more intuitive way to annotate a needle could be line segments. However, for manual annotation, a box was an easy and efficient shape to use. It is also the shape that is commonly used [22,34]. Second, the deep learning algorithms developed in this study were also using a rectangle to annotate the image. Third, the bounding box is naturally paired with a standard evaluation metric, i.e. IoU. There are limitations to using a bounding box as it does not incorporate only the needle but the background as well. Furthermore, a rectangular shape cannot provide information on the orientation and angle of the needle compared to a shape that follows the edge of the needle perfectly.

The agreement between the annotators was relatively strong. However, to improve inter-observer reliability between authors, a strict guideline to annotations could be developed. It is often unclear when the needle should be annotated. By making a stricter set of rules for annotation there will be less variation in the annotations. In theory, this should also improve the effectiveness of the algorithm as the training data given would be more consistent.

The Faster R-CNN has better accuracy but the computational speed is slower than YOLOv3. Both of the algorithms have performance lower than humans. One possible reason for this can be seen in Fig. 11. This is one of the figures that was fed to the algorithm which depicts a blurred needle with many needle looking incisions in the background. There could be various ways to improve the performance of the algorithm. For instance, currently the neural network is given raw images. These images could be pre-processed, e.g. enhanced, for a better performance. Furthermore, other deep learning systems are usually trained with more than 10,000 images. The amount of data in the present application is limited. A boost in performance would be expected if more annotated images were used for training. Also, transfer learning, which initializes the weights of the neural network using the weights trained from other tasks, can be used to improve the algorithm as well.

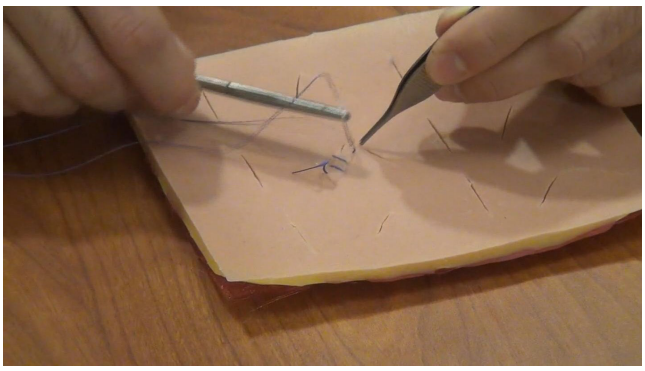


Fig. 11. Sample image of a blurry needle.

5. Conclusion and Future Work

Surgical simulation is becoming an increasingly important component of health care education and now represents a large part of the medical student curriculum. One of the first surgical procedures a medical student learns to perform is suturing. In order for the medical students to improve their skills, they need individualized feedback regarding the position of their needle during the procedure.

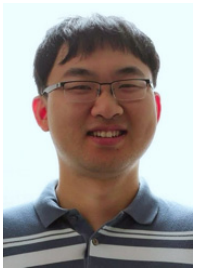
In this study, we implemented two deep learning algorithms to identify the location of the needle on each frame of a surgical video. We compared the performance of the algorithms against the gold standard which is manual annotation using the concept of IoU and two-norm distances between bounding box centroids. The average IoU value of the YOLOv3 algorithm was found to be 0.349, this value was 0.601 for Faster R-CNN, and the average inter-observer value was 0.663. The average two-norm distances between bounding box centroids for the authors and for YOLOv3 and Faster R-CNN were 21.9, 108.2 and 36.8, respectively.

The results are encouraging given the relatively small training dataset. By feeding the algorithm with more test images and preprocessing the images, the performance is expected to improve.

References

1. M. Uemura, M. Tomikawa, R. Kumashiro, T. Miao, R. Souzaki, S. Ieiri, K. Ohuchida, A. T. Lefor and M. Hashizume, Analysis of hand motion differentiates expert and novice surgeons, *J. Surg. Res.* **188** (2014) 8–13.
2. R. Reznick, G. Regehr, H. Macrae, J. Martin and W. McCulloch, Testing technical skill via an innovative bench station examination, *Am. J. Surg.* (1997) 226.
3. J. A. Martin, G. Regehr, R. Reznick, H. MacRae, J. Murnaghan, C. Hutchison and M. Brown, Objective structured assessment of technical skill (OSATS) for surgical residents, *Br. J. Surg.* **84** (1997) 273–278.
4. A. Zia, Y. Sharma, V. Bettadapura, E. L. Sarin, T. Ploetz, M. A. Clements and I. Essa, Automated video-based assessment of surgical skills for training and evaluation in medical schools, *Int. J. Compu. Assisted Radiol. Surg.* **11** (2016) 1623–1636.
5. A. Zia and I. Essa, Automated surgical skill assessment in RMIS training. *Int. J. Compu. Assisted Radiol. Surg.* **13** (2018) 731–739.
6. K. Harada, A. Morita, Y. Minakawa, Y. M. Baek, S. Sora, N. Sugita, T. Kimura, R. Tanikawa, T. Ishikawa and M. Mitsuishi, Assessing microneurosurgical skill with medico-engineering technology, *World Neurosurg.* **84** (2015) 964–971.
7. K. Harada, Y. Minakawa, Y. Baek, Y. Kozuka, S. Sora, A. Morita, N. Sugita and M. Mitsuishi, Microsurgical skill assessment: Toward skill-based surgical robotic control, in *2011 Annual Int. Conf. IEEE Eng. Med. Biol. Soc.* (IEEE, 2011), pp. 6700–6703.
8. N. K. Francis, G. B. Hanna and A. Cuschieri, The performance of master surgeons on the Advanced Dundee Endoscopic Psychomotor tester: Contrast validity study, *Arch. Surg.* (1960), 841.
9. M. A. Farcas, M. O. N. Trudeau, A. Nasr, J. T. Gerstle, B. Carrillo and G. Azzie, Analysis of motion in laparoscopy: The deconstruction of an intra-corporeal suturing task, *Surg. Endosc.* **31** (2017) 3130–3139.
10. J. B. Dimick and O. A. Varban, Surgical video analysis: An emerging tool for improving surgeon performance, *BMJ Quality Safety*, **24** (2015) 490–491.
11. P. Sánchez-González, I. Oropesa and E. J. Gómez, Minimally invasive surgical video analysis: A powerful tool for surgical training and navigation, *Studies Health Technol. Inform.* **190** (2013) 33–35.
12. M. Kranzfelder, A. Schneider, A. Fiolka, E. Schwan, S. Gillen, D. Wilhelm, R. Schirren, S. Reiser, B. Jensen and H. Feussner, Real-time instrument detection in minimally invasive surgery using radiofrequency identification technology, *J. Surg. Res.* **185** (2013) 704–710.
13. S. Speidel *et al.*, Automatic classification of minimally invasive instruments based on endoscopic image sequences, *SPIE Medical Imaging* (2009), p. 72610A.
14. 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, in *2018 IEEE Winter Conf. Applications of Computer Vision (WACV)* (IEEE, 2018), pp. 691–699.
15. A. Vardazaryan, D. Mutter, J. Marescaux and N. Padoy, Weakly-supervised learning for tool localization in laparoscopic videos, (2018).
16. M. Sahu, A. Mukhopadhyay, A. Szengel and S. Zachow, Tool and Phase recognition using contextual CNN features (2016).
17. A. P. Twinanda, D. Mutter, J. Marescaux, M. de Mathelin and N. Padoy, Single- and multi-task architectures for tool presence detection challenge at M2CAI, (2016).
18. T. Liu and M. C. Çavuşoğlu, Optimal needle grasp selection for automatic execution of suturing tasks in robotic minimally invasive surgery, *IEEE Int. Conf. Robotics Automation (ICRA)* (2015), pp. 2894–2900.
19. R. C. Jackson and M. C. Çavuşoğlu, Needle path planning for autonomous robotic surgical suturing, in *2013 IEEE Int. Conf. Robotics and Automation* (IEEE, 2013), pp. 1669–1675.
20. R. Girshick, J. Donahue, T. Darrell and J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in *Proc. IEEE Conf. Computer Vision Pattern Recognition* (2014), pp. 580–587.
21. R. Girshick, Fast r-cnn, in *Proc. IEEE Int. Conf. Computer Vision*, (2015), pp. 1440–1448.
22. S. Ren, K. He, R. Girshick and J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, in *Adv. Neural Inf. Process. Sys.* (2015), pp. 91–99.
23. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu and A. C. Berg, Ssd: Single shot multibox detector, in *European Conf. Computer Vision* (Springer, 2016), pp. 21–37.
24. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, You only look once: Unified, real-time object detection. in: *Proc. IEEE Conf. Computer Vision Pattern Recognition* (2016), pp. 779–788.
25. R. Girshick, Fast R-CNN, in *2015 IEEE Int. Conf. Computer Vision (ICCV)* (2015), pp. 1440–1448.
26. Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, Gradient-based learning applied to document recognition, *Proc. IEEE* (1998) **86** 2278–2324.
27. K. He, G. Gkioxari, P. Dollár and R. Girshick, Mask r-cnn, in *Proc. IEEE Int. Con. Computer Vision* (2017), pp. 2961–2969.
28. K. He, X. Zhang, S. Ren, and J. Sun, Deep residual learning for image recognition, in *Proc. IEEE Conf. Computer Vision Pattern Recognition* (2016), pp. 770–778.

29. A. Kathuria, What's new in YOLO v3? (2018). Available at: https://towardsdatascience.com/yolo-v3-object-detection-53fb7d3bfe6b. Date accessed: 24 September 2019.
30. J. Redmon and A. Farhadi, Yolov3: An incremental improvement. arXiv:1804.02767.
31. Z.-Q. Zhao, P. Zheng, S.-T. Xu and X. Wu, Object detection with deep learning: A review, *IEEE Trans. Neural Netw. Learn. Syst.* (2019).
32. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in *2009 IEEE Conference On Computer Vision and Pattern Recognition* (IEEE, 2009), pp. 248–255.
33. Y. Gao, S. S. Vedula, C. E. Reiley, N. Ahmidi, B. Varadarajan, H. C. Lin, L. Tao, L. Zappella, B. Béjar and D. D. Yuh, Jhu-isi gesture and skill assessment working set (jigsaws): A surgical activity dataset for human motion modeling.
34. J. Redmon and A. Farhadi, Yolov3: An incremental improvement, arXiv:1804.02767.



Qipei Mei is working towards his Ph.D. degree in Structural Engineering at the University of Alberta, Edmonton, AB. He received an MSc degree in Structural Engineering from the same university in 2014. Also, he received an MSc degree in Computer Science from Georgia Institute of Technology, Atlanta, GA, in 2018 and the B.E. degree in Civil Engineering from the Huazhong University of Science and Technology, Wuhan, China in 2011.

Jonathan Chainey completed his medical degree in 2015 at the Université de Montréal, QC. He is now a post-graduate year 5 in Neurosurgery at the University of Alberta and is currently working on his MSc degree in Surgery with specialization in Surgical Education at the same institution.

David Asgar-Deen received his BSc degree in Electrical Engineering from the University of Alberta, Canada, and is working on his MSc degree in Electrical and Computer Engineering (Biomedical) from the University of Alberta, Canada. He is currently acting as a student mentor for the IEEE University of Alberta Student Branch in Canada.

Dr. Daniel Aalto is an Assistant Professor in the Faculty of Rehabilitation Medicine at the University of Alberta. He holds joint appointment at the Institute for Reconstructive Sciences in Medicine (iRSM) where he is a Research Scientist. He obtained his MSc in Engineering Physics from the Helsinki University of Technology in 2005 and his Ph.D. in Mathematics from Aalto University in 2010.