Deep learning for action and event detection in endoscopic videos for robotic assisted laparoscopy

Francis Kaping'a
School of Engineering, Computing and Mathematics
Oxford Brookes University
Oxfordshire, United Kingdom
17037571@brookes.ac.uk

***Abstract*—This work focused on application of deep learning to endoscopic videos in laparoscopy for action detection. The work entailed construction of laparoscopy dataset from scratch, and training and testing a deep learning SSD based online real-time codebase on the new dataset. Deep learning is popular for image and object recognition. Now, it has also consistently gained prevalence in action detection because of promised its advantages and its robustness. Researchers have also committed to explorations of how machine learning can be applied to surgical operations in general and in laparoscopy. Current reviewed works however have performance shortcomings that inhibit their wide adoption in real-world problems such as laparoscopy. Furthermore, reviewed works are customary designed to solve specific problems other than action detection in laparoscopy. We fill the gap by customising a generic online real-time action detection code to explore application of deep learning to laparoscopy. To assess the performance of the model, different approaches were taken with different parameter lists and varied number of training iterations. The exploration reviewed that different approaches yielded different results. The action detection architecture used, and approaches taken yielded high frame AP scores indicating that deep learning could be applied in in laparoscopy. The diverse dataset constricted contained 35 action classes that different durations. 27 classes scored more 80% frame AP**

Keywords— Laparoscopy, Convolutional Neural Networks, Machine learning, Action detection, Real-time action detection, Supervised learning, robotic-assisted surgery.

I. INTRODUCTION

Laparoscopy, also referred to as minimally invasive surgery (MIS), is a type of surgical procedure that allows a surgeon to access the inside of the abdomen and pelvis without making large incisions [15]. Its advantages include quick healing, less pain, and lower chances of infections compared to open surgery. It has hence been explored for wide applications in the medical field. With advent of improved technology, robotic-assisted laparoscopy has seen increased use over the years. Robotic Assisted MIS entails having trocars with instruments mounted on them inserted into the abdomen of the patient. These instruments are then controlled from consoles by the main operating surgeon and the assistant operator.

The extra staffing requirement in current setting has its draw backs. First, it is costly. Second, it is delegates too much work to humans who are prone to make mistakes. Third, the main operator and the assistant operator, not looking at the same console, have communication mishaps leading to assistant not doing what the main surgeon would have required of them. For these and other reasons, the Smart Autonomous Robotic Assistant Surgeon (SARAS) consortium have embarked on developing the next-generation of surgical robotic systems that will allow a single surgeon to execute Robotic Minimally Invasive Surgery (R-MIS) without the need of an expert assistant surgeon [16]

To contribute to the ongoing SARAS project, this work aimed to first construct a new laparoscopy dataset by annotating a endoscopic video by drawing bounding boxes around actions of interest in video frames. Second, adapt code provided by Singh et al [14][17] to the new dataset. Lastly, train and test the resulting code on the new dataset.

R-MIS has seen increased use especially on deformable anatomical structures [18]. Its wide adoption and experienced benefits have attracted substantial interest from academia. Many have explored equipping of the robot involved with various aspects of intelligence to leverage computing power and further increase the benefits of the involved robot. Furthermore, many researchers have done several works in machine learning and action detection as it will be seen in the proceeding section. While many inform our literature to a good extent, they do not focus on the problem being investigated in this work. Most of computer vision work done has been based on image or object detection, classification or regression. The surveyed action detection works recommend architectures that either work offline or have performance shortcomings that inhibit their adoption. Building upon a fast network architecture [13], Singh et al built a novel online architecture that scored good results on benchmark UCF101-24 dataset [14].

To the best of my knowledge, this is the first work that takes advantage of a real-time action detection architecture built on real-time Single Shot Detector to investigate the application of deep learning in laparoscopy.

II. RELATED WORK

A. Action detection

Action detection is a step above image of object detection in that detected objects will have to be considered over a span of frames to detect actions. This area has gained interest from academia. For example, Kuo et al explored an approach of using a pre-processing procedure to extract key-frames from a video sequence and provide a compact representation for the video [1]. In their system, they represented action objects using mixture of deformable patch models. The patches within the bounding boxes were then sampled to

extract their spatial and temporal features to train the deformable models. Another technique explored is Generalised Hough Transform (GHT) which uses template matching [2]. GHT generally refers to any detection process based on additive aggregation of evidence (Hough votes) coming from local image/video elements [3]. Gall et al performed an extensive work in action recognition by making use of codebooks to classify the local appearance of interest points into a discrete number of visual words that represent an object class. Hough transform was used to detect objects, and Hough forests were used as decision trees learned on the training data [3]. Each tree in the Hough forest maps local appearance of image or video elements to its leaves, where each leaf is attributed a probabilistic vote in the Hough space [3]. They reported impressive classification results of 95.6% and 92.0% for Weizmann and KTH datasets, respectively.

B. Machine learning in laparoscopy

With continue proliferation of applications of machine learning, it has also found its use in the health domain. Several researchers have explored techniques of equipping computers in the health sector with relevant intelligence. Very similar to this work is that of Choi et al [4] that explored the use of Convolutional Neural Networks (CNN), designed based on YOLO (You Only Look Once), in recognition of surgical instruments used in minimally invasive robotic assisted surgical procedures. YOLO methodology in their architecture allows real-time detection of surgical instruments which is desirable. They reported a good mAP of 72.26%. Their model however is restricted to object detection which is insufficient for the action detection work. Another closely related work is by Voros and Hager [6]. They used Naïve Bayes to determine which of two classes an event belonged to. The classes considered were ‘’interaction” and ‘’no interaction”. This was to train the model and later regress locations in data where tools interacted with tissue in a real-time manner. Their model was quite simple and attained a good accuracy of 81%. [6] differs from our work in two major ways: first, they only considered two generic classes. Second, Naïve Bayes while nicely simplistic, strongly relies on training data during test. It calculates probabilities of the given data belonging to the available classes based on data in training set. This inhibts its application to real-world problem with huge datasets.

Despinoy et al [5] endeavoured to develop a method for the automatic and quantitative assessment of surgical gestures. There model was only for training surgery staff and not applicable to the problem explored to this work. Another exploration of application in robotic assisted laparoscopy is by [9] and [10]. [9] studied and demonstrated the use of CNNs to detect and track tips of tools used in robotic minimally invasive surgery. There models yielded good results but they are restricted to object and image detection.

C. Real-time action detection

One of the drawbacks of linear classifiers is the requirement for plenteous data and several training iterations to arrive at an optimal hyperplane. Jung et al [7] countered this problem by proposing a method of building strong action classifiers from a small video dataset that is based on a representative example. Similarly, Malawski and Kwolek [8] proposed a method of predicting actions by not entirely relying on feature extraction. They nominate action candidates by analysing actors’ velocity using skeleton data provided by a Kinect. They also hugely relied on segmenting data and gaining confidence about following action based on maxima identification. [7] however failed to perform when actions happened fast. [8] was designed and only tested on a small niche dataset. Its high accuracy was attained because all data involved only belonged to one class.

With an aim to predict and localise actions in videos, Gkioxari and Malik [11] selected most prospective regions in frames and used CNNs to classify them. To encode spatiality of actions, [11] used an approach of having a second pass through the data to link frames and construct “action tubes”. The second pass takes the model far from real-time architecture. To turn from this inefficiency, Saha et al [12] proposed an end-to-end trainable model which considers 3-dimentional regional proposals; two consecutive frames. This novelty allowed linking of frame in the first pass, consequently not needing the second linking pass. Drawing from [12], Saha et al built an action tubes detection architecture that built time-spaced action tubes [19].

Further going towards real-time object detection, [13] presented a deep network-based object detector that does not resample pixels or features for bounding box hypotheses. Their method, referred to as The Single Shot Detector (SSD), was explored further by Singh et al and extended to an online real-time multiple spatiotemperal action localisation and prediction architecture [14]. Building upon [19] and leveraging the SSD architecture’s high performance and accuracy for object detection [13], Saha et al devised an algorithm that fuses detections from appearance and flow SSD networks and later incrementally builds action tubes. Their method yielded excellent results on a very diverse dataset; UCF102-24. This novel method is therefore the pedestal for work in this project.

III. METHODOLOGY

A. Convolutional Neural Networks

The notion of Neural Networks finds its history and inspiration in animal brain neurons [20]. It is based on a collection of nodes that loosely simulate the functioning of animal brain neurons in pattern recognition patterns. The nodes are interconnected in a fashion of a multi-layered perceptron, with information flowing from input nodes to output nodes in one way.

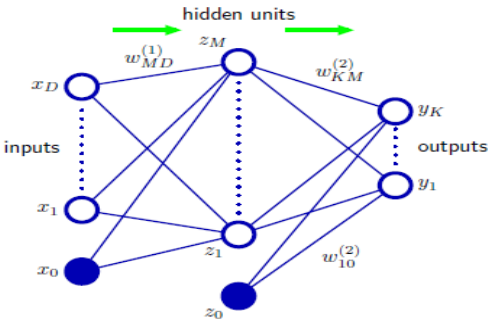

Figure 1: Source [21]

The nonlinear activation function is evaluated on given input to determine which node to activate. Before the activation function is applied, input values are transformed [21]. CNNs are a flavour of ANNs that have popularity in the domain of image recognition. CNNs maintain an invariance through the network layers [21] and their good performance is aided by having layers in the shape of an input image. Through max pooling and down sampling, CNN reduces the amount of data being processed, thereby improving performance. CNN based architecture then becomes a natural choice for image processing in laparoscopy.

B. *Real-time online architecture*

SSD uses VGG-16 architecture-based network for image prediction. Its real-time performance comes from its design that allows it to perform both object localisation and classification in a single pass forward instead of requiring a pass for each [13]. SSD also abandons numerous bounding box proposals and pixel sampling stages to make the model much faster [13]. Leveraging SSD real-time architecture, Singh et al [14] built a real-time online action detection model. The architecture built in [14] is the foundation for in this exploration project. The model receives as input a sequence of RGB images. The RGB frames are sent to Appearance SSD for appearance scores computation. The RGB frames are also sent to Flow SSD for flow scores computation. The two parallel predictions are then fused and in the last stage the action tubes are built incrementally.

C. *Data preparation*

To create the dataset from scratch, data annotation was done using Visual object Tagging Tool (VoTT); a Microsoft open source tool used for drawing bounding boxes around regions of interest in visual data. In our case, it was used to draw bounding boxes around regions with actions of interest in video frames. Annotation rate was 4 frames per second. This resulted in 35 action classes and 23558 frames which sufficed for the proof of concept project.

While annotating the data, decisions were made based on machine learning knowledge. To begin with, what constituted an event was not clear. Some researchers have considered surgical tools tracking methodologies [9] but that used in action detection and classification would evidently cause overreliance of model on tools. The model will detect many false actions whenever a tool comes into vicinity. Basing actions on organs or tissues being within vicinity has similar shortcomings; the model will detect false actions whenever it sees tissues or organs. This work therefore explores the use of a combination of both organs and tools. Bounding boxes were drawn only when tools were on the appropriate organs for actions of interest. Furthermore, there was a question of the ideal sizes of the bounding boxes. To balance the presence of tools and organs or tissue in a bounding box, bounding boxes were restricted to containing 30%-70% of either tools or organs. This was for purposes of lessening sensitivity of the model to the either tools or organs to encourage correct predictions. Thirdly, a decision was made to only begin an action when a tool was close enough to the appropriate organ. Indicating an action when a tool is far from appropriate organ can potentially lead to a misleading dataset as there are many video segments when tools appear around organs but with negative actions. The afore mentioned dilemmas constituted challenges of developing a dataset good enough for laparoscopy.

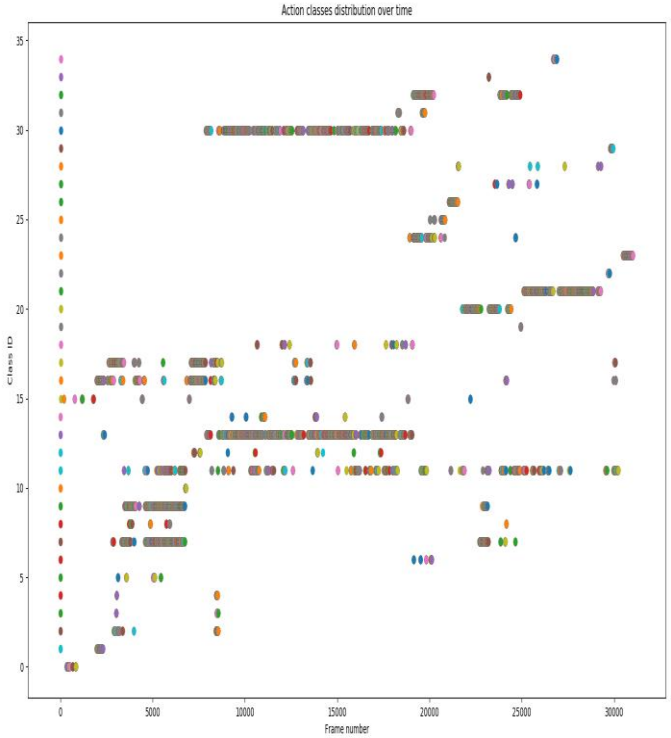

Figure 2: *Distribution of actions over time*

D. *Data segmentation*

As observed from the diagram below, some classes spanned longer portions of the video while others happened in short intervals. Again, others were distributed over several regions of the video while others only occurred either at the beginning, in the middle, or at the end. Simply segmenting the video into two portions (training and testing) would result in many action instances being available only in either training or testing. To reduce this problem, two approaches were taken. In approach 1, conceptually 65% of

the video was considered training data and the rest test data. The video was then split into 10 portions rather than 2 portions. Each of the 10 portions was then split into training and testing data. In approach 2, 75% of the video was used for training and 25% for testing. The video was split into 38 portions of 800 frames each. Within each region, 600 frames were used for testing and 200 for testing. These two approaches endeavored to ensure that as many instances were captured and made to have both training and testing data. Results for these approaches are discussed in section V.

IV. EXPERIMENTS AND RESULTS

This section gives an overview of the results from different experiments that were conducted. Each experiment had minor changes to value of IoU, number of iterations, and the way the data was organised. As observed below, different approaches yielded different results.

A. Approach 1 experiment and results

In this experiment, IoU was set to 0.2 and the total number of iterations was set to 120,000. Data was split according to approach 1. In this approach, while the problems mentioned in III C were minimized, we still had 15 of 35 instances that had either training or testing data missing. This was because the portions the video was split into were large enough to possibly have instances of a class gathered on one side (either training or testing). Results for such classes are guaranteed zeros and therefore do not represent the performance of the model. There are considered 'bad data'.

The results from approach 1 experiment were not very good. Excluding bad data, the mean frame AP was 27.12%. Including bad classes, the score is 13.95%. In this experiment, only a few classes exhibited good performance. The diagram below shows per results obtained from this experiment.

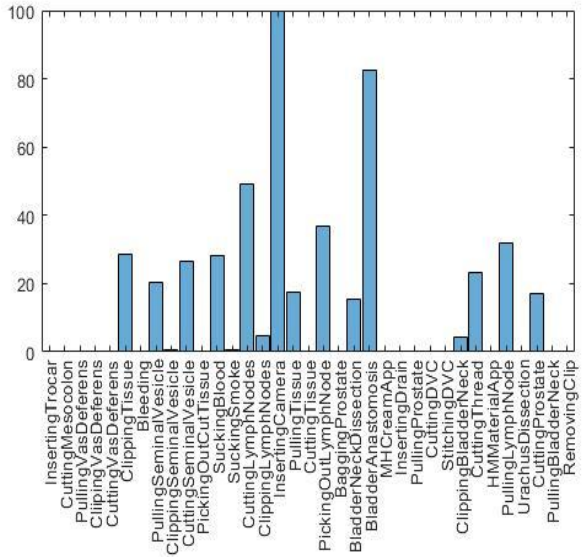

Figure 3: *Approach one performance*

B. Approach 2 experiments and results

In this experiment, IoU was set to 0.5, and the total number of iterations was 150,000. Furthermore, the data was split according to approach. In this experiment, 24,229 frames were extracted from the video. The training data summed up to 17400 frames while testing data 6158 frames. The total number of boxes was 35003. In approach 2, the problems highlighted in III. C were alleviated to a larger extent, leaving only 5 classes affected. 15 experiments were carried out for this approach, each having a different number of iterations. As seen in figure 5, the performance improved as the number of interactions increased. The diagram below shows results from experiment 15 with 150,000 training iterations.

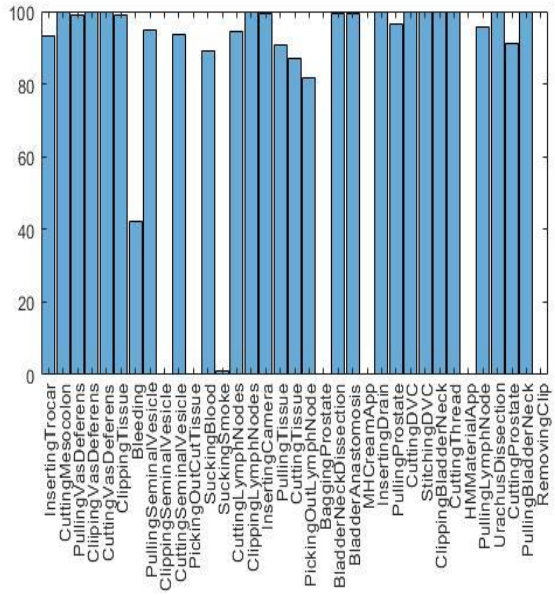

Figure 4: *Approach two performance*

In the 15 experiments, performance was also tacked against the number of iterations. It was noted that after 40,000 training interactions, the model was not able to improve any further as indicated in the diagram below.

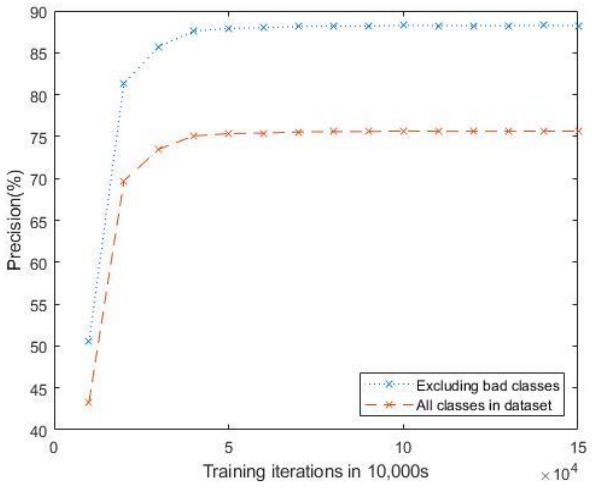

Figure 5. *Precision over performance.*

V. DISCUSSION

A. *Impact of data segmentation*

As mentioned in the preceding section, some classes had no training data and some others do not had testing data. This was because data was split in time rather than just ensuring that each class had sufficient representation in both training and testing sets. This complexity was introduced by possibility of having several classes on one frame. Copying frames to class folders results in duplication of frames. This is a problem because one frame would be seen as positive for one class and negative for another class which may affect convergence and lead to skewed results. Furthermore, a frame could possibly be on training end for class 1 and be on testing end for class 2. This also could skew results. The two approaches explained in section III are the most appropriate for splitting one video into training and testing data. As seem from figure 3 and figure 4, approach two yielded much better results. Less instances were affected and there was a better distribution of training and testing data.

B. *Impact of training iterations and IoU*

The model in approach one was trained with 120,000 iterations. It was noted notes that beyond 40,000 of iterations the model did not improve any further. This was concluded by observing the loss function over several number of iterations. The IoU used approach one was 0.2. The model scored 27.12% after 120,000 of training iterations. The results obtained in this approach prompted a significant shift in data organisation and the parameters that were used.

Experiments in approach two yielded much better results compared to approach one. In this approach IoU was set to 0.5 and number of maximum iterations increased to 150,000. Just like approach one, results were more impressive with higher number of training iterations. Experiment 2.10 was conducted with a model that was trained with 10,000 iterations only. As observed from that experiment, the average precision was 43.33% with 15 classes scoring below 20% accuracy. Excluding the bad classes, the average was 50.56%. Of the total number of action classes, 6 scored above 80% while the rest were in between 20% and 80%. At this stage, clearly training session had not yet produced a model good enough for new predictions.

Training iterations were increased in increments of 10,000 until the performance was observed not to improve any further for all classes. As seen in figure 5, after 40,000 training iterations, almost all classes maintained a constant score indicating that further iterations were not needed, just as observed in approach one. ‘Bleeding’ action maintained score of around 40%. ‘Clipping Seminal Vesicle’ and ‘Sucking Smoke’ actions maintained scores below were below 2%. Other classes maintained a score above 80%.

C. *Data sufficiency*

Another performance driving factor was availability of training and testing data. Enough training data ensures that the model converges to a set of weights that cover enough types of classes data. Enough testing data ensures the model is tested against sufficient variations of the data belonging to a class to give a more reliable accuracy score. Classes with enough training and testing examples had a good smoothed score over all testing rounds. Classes such as ‘Sucking Smoke’ with 140 training and 4 testing examples, and ‘Clipping Seminal Vesicle’ with 95 training and 23 testing examples yielded the worst results. Arguably, minimal training and testing data can also lead to misleadingly impressive results. For example, it is very easy for the model to score right on all 10 testing examples especially if the variation among the data is not enough. It is also easy for the model to get all the 10 tests wrong especially if the 10 are perceived different from the weights learnt from the training data. Some classes yielded very impressive results for nearly all iterations. However, it can be observed that none of the classes that scored 100% on several tests had more than 200 examples in both training and testing sets. Those sufficiently represented had scores in the range 90% < score < 100%. Having enough training and testing data therefore is what gives confidence in results yielded.

D. *Comparison of ground truth to predictions*

The model performed very well on bounding box predictions. Most of the bounding boxes had the value of IoU close to 1. As expected, there were a few with bad IoU values. The following figure show ground truth boxes against those predicted.

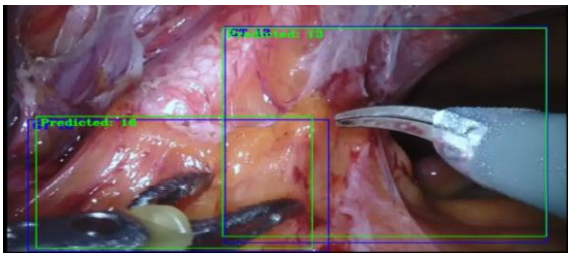

Figure 6: *Ground truth and predicted boxes*

VI. CONCLUSIONS

Action detection has gained attention in academia because of its promised benefits. There are many methodologies that have been explored as indicated in the literature review. However, approaches explored had their shortcomings and could not be applied to the detection of actions in laparoscopy. Most of them had performance shortcomings while others were made for other specific problems. This project therefore aimed at filling the gap by first preparing a laparoscopy dataset from a video recorded using an endoscopic camera from a robotic assisted surgical operation.

Due to time constraints, only two approaches were explored. In approach 1 where the video was split into 10 portions, model trained at 120,000 iterations and IoU set to 0.2 27.12% precision was achieved. The score was low enough to prompt a significant shift in the approach. Approach 2 had training iterations set to 150,000, IoU set to 0.5, and the video split into 38 portions. Of 35 classes, approach 2 yielded more than 80% score for 27 classes. In fact, most of the classes were above 90%.

The results obtained in this project reviewed that deep learning can possibly be used in laparoscopy. To gain more confidence in this conclusion however, extending this work to perform video AP will be vital. The results presented in this paper are limited to frame level detections. The good results obtained in approach two however are a good indication that video AP will likely be impressive. This conclusion logically makes sense because video AP is based on aggregation of frame level detections, which have been proved in this work to be excellent.

REFERENCES

[1]D. Kuo, G. Cheng, S. Cheng and S. Lee, "Detecting Salient Fragments for Video Human Action Detection and Recognition Using an Associative Memory", *2012 International Symposium on Communications and Information Technologies (ISCIT)*, pp. 1039 - 1044, 2018.

[2]"Generalised Hough transform", *En.wikipedia.org*, 2018. [Online]. Available: https://en.wikipedia.org/wiki/Generalised_Hough_transform. [Accessed: 26- Sep- 2018].

[3]J. Gall, A. Yao, N. Razavi, L. Van Gool and V. Lempitsky, "Hough Forests for Object Detection, Tracking, and Action Recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 11, pp. 2188-2202, 2011.

[4]B. Choi, K. Jo, S. Choi and J. Coi, "Surgical-tools Detection based on Convolutional Neural Network in Laparoscopic Robot-assisted Surgery", *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 1756 - 1759, 2017.

[5]F. Despinoy, D. Bouget, G. Forestier, C. Penet, N. Zemiti, P. Poignet and P. Jannin, "Unsupervised Trajectory Segmentation for Surgical Gesture Recognition in Robotic Training", *IEEE Transactions on Biomedical Engineering*, vol. 63, no. 6, pp. 1280-1291, 2016.

[6]S. Voros and G. Hager, "Towards "Real-time" Tool-tissue Interaction Detection in Robotically Assisted Laparoscopy", *2008 2nd IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics*, pp. 562 - 567, 2006.

[7]S. Jung, Y. Guo, H. Sawhney and R. Kumar, "Action exemplar based real-time action detection", *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, 2009.

[8]F. Malawski and B. Kwolek, "Real-Time action detection and analysis in fencing footwork", *2017 40th International Conference on Telecommunications and Signal Processing (TSP)*, 2017.

[9] Z. Chen, Z. Zhao and X. Cheng, "Surgical instruments tracking based on deep learning with lines detection and spatio-temporal context", *2017 Chinese Automation Congress (CAC)*, pp. 2711 - 2714, 2017.

[10]S. Wang, A. Raju and J. Huang, "deep learning based multi-label classification for surgical tool Presence detection in laparoscopic videos", *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*, pp. Pages 620-623, 2017.

[11]G. Gkioxari and J. Malik, "Finding action tubes", *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[12]S. Saha, G. Singh and F. Cuzzolin, "AMTnet: Action-Micro-Tube Regression by End-to-end Trainable Deep Architecture", *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017.

[13]W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu and A. Berg, "SSD: Single ShotMulti Box Detector", 2018.

[14]G. Singh, S. Saha, M. Sapienza, P. Torr and F. Cuzzolin, "Online Real-Time Multiple Spatiotemporal Action Localisation and Prediction", *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017.

[15]"Laparoscopy (keyhole surgery)", *nhs.uk*, 2018. [Online]. Available: https://www.nhs.uk/conditions/laparoscopy/. [Accessed: 26- Sep- 2018].

[16]"Project – Saras Project", *Saras-project.eu*, 2018. [Online]. Available: https://saras-project.eu/?page_id=13. [Accessed: 26- Sep- 2018].

[17]G. Singh, "gurkirt/realtime-action-detection", *GitHub*, 2018. [Online]. Available: https://github.com/gurkirt/realtime-action-detection. [Accessed: 26- Sep- 2018].

[18]D. Stoyanov and G. Yang, "Soft tissue deformation tracking for robotic assisted minimally invasive surgery", *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2009.

[19]S. Saha, G. Singh, M. Sapienza, P. Torr and F. Cuzzolin, "Deep Learning for Detecting Multiple Space-Time Action Tubes in Videos", 2018.

[20]"Artificial neural network", *En.wikipedia.org*, 2018. [Online]. Available: https://en.wikipedia.org/wiki/Artificial_neural_network. [Accessed: 26- Sep- 2018].

[21]C. BISHOP, *PATTERN RECOGNITION AND MACHINE LEARNING*. [S.l.]: SPRINGER-VERLAG NEW YORK, 2016, pp. 268 - 271.