Tool Classification in Open Surgery Using a Multi-Camera Video System

My Digital Nurse - Research Project

Naor Haba 206014185

***Abstract*—In the following paper I will present my work during the research project under the *Sensor-Analytics for Clinical Performance Lab* at the Technion. The project is a part of a research about the possibility of creating a digital aiding system, *My Digital Nurse*, which aims to predict the next used tool by the surgeon during an open surgery. The system is based on a multi-camera video system, which records the surgery tools, the surgeon's hands and the assistant's hands from 2 different views - a close up view (side) and an overhead view (top).**

I will present and compare 2 approaches to solve the tool classification task - early fusion and late fusion, regarding the data integration process from the 2 camera views. After determining the leading approach I will also compare the benefit of using 2 camera views to a single camera view. The report is concluded with the leading approach reaching to 86% accuracy on the test set.

I. Introduction

An operating room video recorder is an essential tool for assessing and documenting activity in the operating room and surrounding the surgical site for such purposes as surgical education, phase recognition, workflow analysis, error analysis, skill appraisal, and video summarization. It is true that many environments are able to utilize a single camera to provide the best possible picture, however, in open surgery multiple cameras are required to give a comprehensive picture due to the dynamic nature of the working environment as well as the need to identify different anatomical landmarks and distinguish between tools that are similar. There has been a lot of research in the field of computer vision regarding the concept of object detection, and many methods have been developed in order to solve this task in both an efficient and effective manner. Studies have also found that methodologies based on multiple input sources have shown promising results in recent studies and have become increasingly popular. It is common to refer to this process as ”fusion” when discussing methods that utilize multiple camera inputs, as it describes the process of combining the information from these sources in order to create one comprehensive representation of the data. There are two stages of the fusion process that need to be distinguished when discussing it. These are early fusion and late fusion. Early fusion is done by combining the input data before the feature extraction process, while late fusion is done after the feature extraction process. In this work I will explore both approaches. In the early fusion

approach I will train a transformer based network that combines the information from both inputs in a local manner (among views) and then in a global manner (between views) - based on Chen et al's work [1]. The information is than processed in a time-series network also based on transformers architecture GPT2 [2]. In the late fusion approach I try a similar approach to Kristina's work [3] by training YOLOv7 [4], an object recognition network, on the task of recognizing tools in the images and then using it as a feature extraction network for the images. I then train a classifier network that combines the features from the 2 views and process them in a time-series network using temporal convolutions based on MSTCN [5].

II. Methods

A. Data

In order to be able to collect data for the training of our system, we had participants (surgeons) perform a surgical assessment using an interactive simulator that represented a trauma patient in the operating room with an injured bowel. They had 15 minutes to complete the task and were assessed on the quality of the repair. A camera was fixed on a frame to collect video data from above, and a second camera was set up as a zoom-in to the surgeon's hands and the intestine. Overall we used 62 videos labeled with bounding boxes (recognition) and 62 videos labeled only with used tools in each hand (classification). The videos are divided into 2 views which leaves us with total of 31 surgeries labeled for recognition and 31 surgeries labeled for classification. The data included 20 classes: 2 surgeon hands and 2 assistant hands, each hand either empty or holding one of 4 tools. To accomplish the classification task we used *global ground truth* which suggested the true label for each time in the surgery as recorded from both of the views.

B. Early Fusion Approach

In early fusion approaches we try to combine the information from several inputs describing the same scene before or during the feature extraction process. This way, the features describing the scene are learned from the combined information of all inputs. The networks used for this approach are usually more complex and deeper than the ones used in late fusion approaches. In this work, for the spatial information processing (processing 2 frames that describe a scene in the same time of the surgery) I use a transformer based network that combines the information from both inputs in a local manner (among views) and then in a global manner (between views) - based on the paper [1]. This local-global manner of processing allows the network to learn the crucial features from each view separately and then learn the global features present in both views into a total representation of the scene. For the time information processing (processing the scene information through the time of the surgery) and to complete a network based on transformers entirely, I use a time-series network based on transformers architecture GPT2 [2]. A diagram describing the

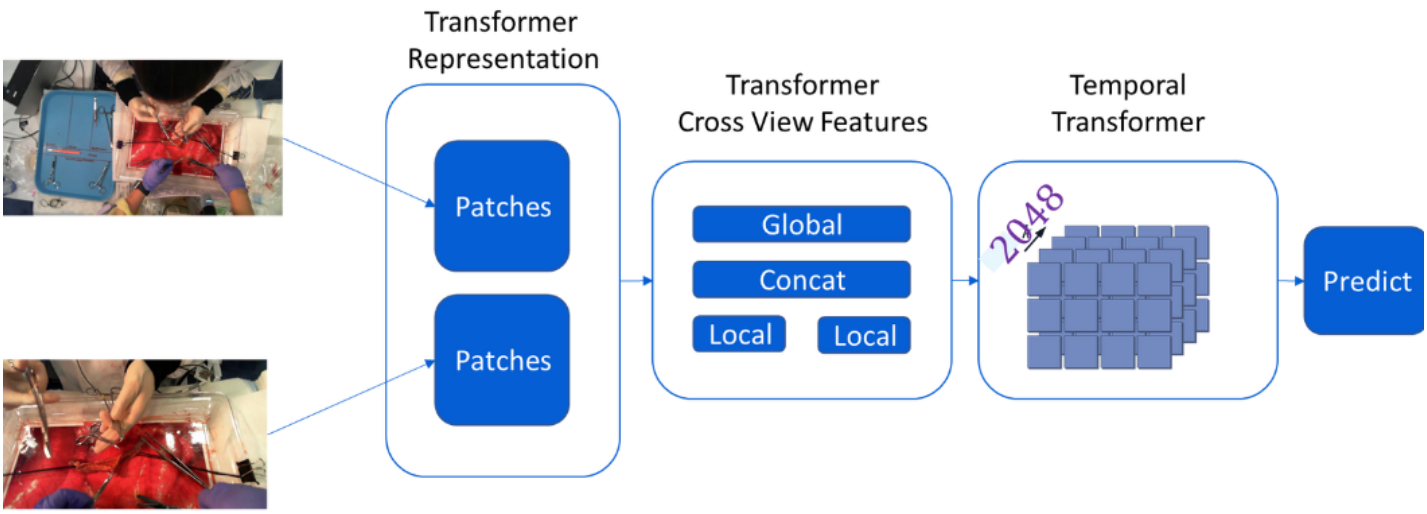

Fig. 1. Early Fusion Approach Diagram

total architecture can be seen in figure 1.

C. Late Fusion Approach

In late fusion approaches we combine the information from several inputs after the feature extraction process. Meaning, we first learn the features from each input separately and then combine them to a single representation of the scene. Usually the feature extraction from each input is done using a pretrained network (fine-tuned or not), which leads to a simpler overall network. The possibility of using a pretrained model is the advantage of such networks, as they are typically more efficient and faster to train. In this work, for the spatial information processing I use a pretrained YOLOv7 [4] network and fine-tune it to recognize the tools in each view independently and their bounding boxes. By using a different task which is more complex than the classification task, I hope to get a better feature extraction network, as this task requires the network to capture information from the entire image and not only from the tools or hands. After training the YOLO network, we use it to get the features for each frame and view. Looking at the architecture of YOLO we can see there are several prediction heads in different places along the network. Each head has a different lookup size of the image and the last head has the broadest lookup and accordingly the least features used before predicting. Due to memory capacity limits, we therefore select the features of an image as the 4500 features extracted through the network and before the last head. For the time information processing, again to complete a network using the same techniques, we use a time-series network based on convolutions in the time domain - MSTCN [5]. A diagram describing the total architecture can be seen in figure 2.

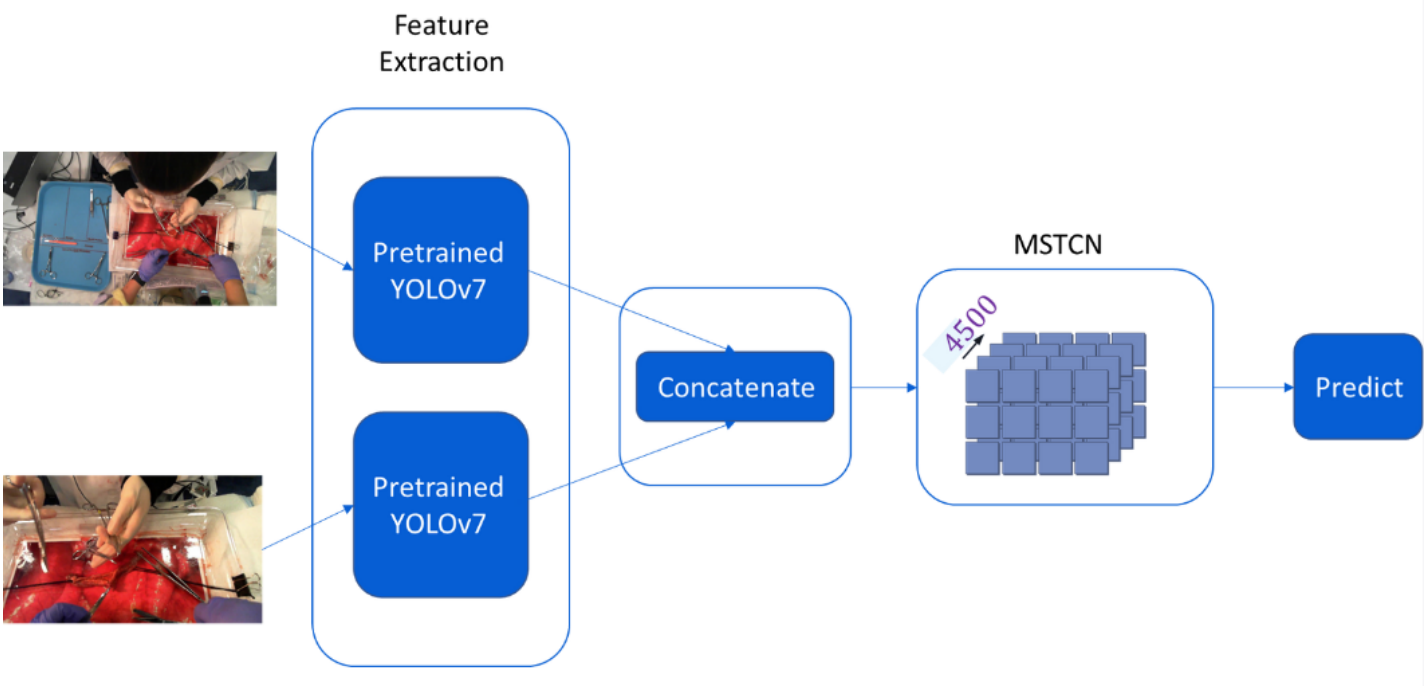

Fig. 2. Late Fusion Approach Diagram

III. Results

A. Early Fusion Approach

A rather disappointing outcome has resulted from the use of this approach. The network was trained using many configurations and hadn't surpassed 13% accuracy on the validation set. We suggest the following assumptions to try and explain this phenomenon:

1) Complex Network - The network is too complex for the task at hand. The network is based on transformers architecture which is known to be very complex and hard to train. The network is also very deep and has a lot of parameters.
2) Lack of Data - The data is not enough to succeed in this sort of task. The network is trained using only 36 surgeries, which is not enough to train a network to solve the task at hand.
3) Views Redundant Information - The network is trained on 2 views of the same surgery, which means that the network is trained on the same data twice. This leads to a lot of redundancy in the data and makes it harder for the network to learn the features.

B. Late Fusion Approach

Unlike the previous approach, this approach has reached a much better result, and can be defined as a success. Training the YOLO network we reached a precision of around 75% by fine-tuning other YOLOv7 model for our task. Then, after performing hyperparameter tuning to find the best configuration for the network, the late-fusion network has surpassed 86% accuracy on the validation set and

reached 86% accuracy on the test set. This result has led me to conclude that the phenomenon of the failure of the early fusion approach is probably the *Complex Network* reason. The early fusion network is too complex for the task. The late fusion approach, on the other hand, has a much simpler network and had also been pretrained by using the state of the art YOLOv7 network. This network is much more efficient and easier to train, and therefore it succeeded in solving the task.

IV. Discussion

Through this paper our aim was to investigate the effect of using two camera views on the tool classification task. As each camera may find different features in the scene and reveal different tools classifications, the problem of integrating the two views of the scene required us to make a choice between two approaches. These approaches are known as early fusion and late fusion, and each of them comes with its own advantages and disadvantages. Examining the results, we can see that the late fusion approach succeeded in solving the task, while the early fusion approach failed. Studying the following aspects reveal some more information regarding the approaches:

A. Using One View Only

Using the better approach, I trained a network using only one view of the surgery to try and understand the effect of fusing the information from 2 views. My assumption was that the network would perform worse than the late fusion network, as it would have less information to learn from. However, as seen in the following table (I), when training the network using the close-up view (side view) we reach similar results to the fusion network. This

	Both Views	Top View	Side View
Train Acc	0.969	0.907	0.955
Validation Acc	0.857	0.77	0.844
Test Acc	0.862	0.81	0.872

TABLE I

result suggests that the close-up view is enough to solve the task, and that the information from the top view is redundant in this case. Looking more closely at several surgeries we see that both the fusion network and the network trained on the close-up view perform well when the tools are clearly visible in the close-up view and ”logically guess” the tools when they are not visible at all. However, when the tools are not visible in the close-up view but are visible in the top view, the network trained on the close-up view sometimes fails to ”guess” the tools. The fusion network also sometimes ”guess” the wrong tool as if it were looking only from top view, meaning, occlusion from top view can result in inaccurate classification from the fusion network. We therefore conclude that the top view is crucial in this task, as it provides information that the close-up view does not.

B. Using YOLO Labels

Another interesting aspect is the effect of using the YOLO labels as the input to the network. In the late fusion approach, we originally use the YOLO features from the last head. However, we can also

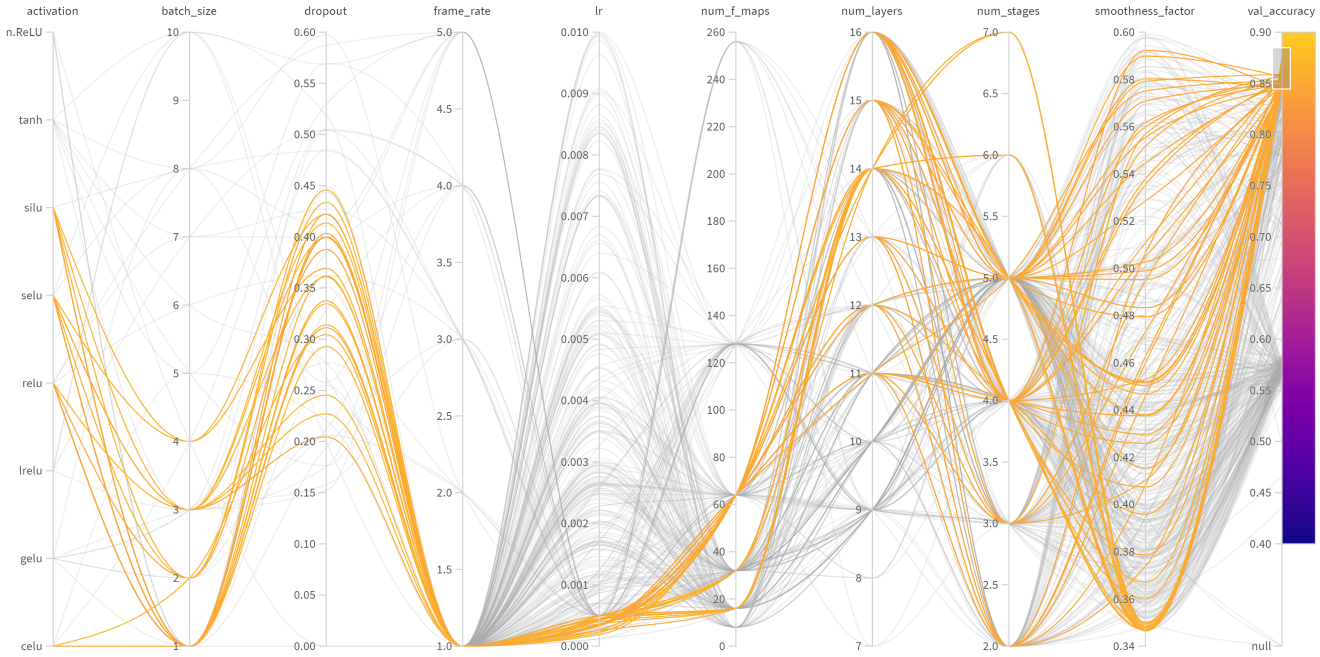

Fig. 3. Hyperparameter Tuning Experiments

use the YOLO labels as the input to the network. This way, we can use the YOLO network to classify hand tools together with their bounding box in each frame and use them as the features to that frame. Specifically, we create a 36 entries vector containing information about the tool used in each hand and its location in the image (5 one-hot encoding entries for the tool classification and 4 entries for the bounding box, all for 4 hands). As we would expect, due to the less informative nature of the YOLO labels, the network trained on the YOLO labels performs much worse than the network trained on the YOLO features. The network saturates very quickly and reaches only 58% accuracy on the validation set. This result suggests that the YOLO features are more informative than the YOLO labels, and therefore we should use the YOLO features as the input to the network.

C. *Hyperparameter Tuning*

Learning from over 1000 experiments done using the late fusion network, we can also reach some interesting conclusions regarding the training process of the network and discover which parameters control the success of the network. By using the graph created in WandB platform and shown in figure 3, we learn the following correlations:

- Batch Size - Since the features are precalculated by the YOLO network, here we refer to the amount of surgeries entering the time-series network. The network performs better when the batch size is lower. This is probably due to the fact that the network is trained on a very small dataset and therefore needs to be

trained slowly to avoid overfitting.

- Frame Rate - This parameter controls the rate of frames we use from the surgery, where 1 is every frame and 10 is every 10th frame. The network performs better when the frame rate lower (more frames). This is rather intuitive, as the network has more data to learn from and this also suggests that there is not much redundancy between the frames.
- Number of Layers - This parameter controls the network view along the time domain. Higher number of layers means a broader view of the time-series network, which means we combine knowledge from more frames along the surgery. As we can see, the network performs better when the number of layers is higher which means that combining knowledge from more frames is beneficial.
- Number of Stages - This parameter controls the amount of refinement stages the prediction goes through. The first stage is always the prediction stage, and the others are used to refine the prediction by using the previous prediction as the input to the next stage. As we can see, the sweet spot for this parameter is 4 stages, which means that the network performs better when it has 3 refinement stages. It seems that too high number of refinement stages makes the network too deep and therefore harder to train, while too low number of refinement stages makes the network too shallow and therefore less accurate.

A deeper dive into the hyperparameter tuning process can show even more interesting correlations, but for the sake of brevity we will not go into them.

V. Future Work

As I mentioned earlier, this project is a part of a research, aimed to create a digital aiding system for the surgeon. Together with another research done by Lior Yariv, we aim to integrate our methods into a single system, which will be able to predict the next used tool by the surgeon. To achieve this goal, future work will have to focus on understanding the different ways of combining the knowledge of surgery scenery learned by the tool-classification system (my network) together with the knowledge of surgery plans learned by Lior's system. The way we see it now, we need to use the combined features from our systems for each point in a surgery and process it further using a recurrent neural network. This will allow us to predict the next used tool by the surgeon, given the current state of the surgery. This is a very interesting and challenging task, which will require a lot of work and research. I hope this research will open new doors in the field of digital aiding systems for surgeons and in the field of systems combining multiple sources of input data.

References

[1] Chen, S., Yu, T., Li, P. (2021). Mvt: Multi-view vision transformer for 3d object recognition. arXiv preprint arXiv:2110.13083.

[2] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I. (2019). Language models are unsupervised multitask learners. OpenAI blog, 1(8), 9.

[3] Basiev, K., Goldbraikh, A., Pugh, C. M., Laufer, S. (2022). Open surgery tool classification and hand utilization using a multi-camera system. International Journal of Computer Assisted Radiology and Surgery, 17(8), 1497-1505.

[4] Wang, C. Y., Bochkovskiy, A., Liao, H. Y. M. (2022). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. arXiv preprint arXiv:2207.02696.

[5] Farha, Y. A., Gall, J. (2019). Ms-tcn: Multi-stage temporal convolutional network for action segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 3575-3584).