Fine-tuning Deep Architectures for Surgical Tool Detection

Aneeq Zia**, Daniel Castro** , and Irfan Essa

Georgia Institute of Technology, Atlanta, GA, USA
http://www.cc.gatech.edu/cpl/projects/deepm2cai

1 Introduction

Understanding surgical workflow has been a key concern of the medical research community. One of the main advantages of surgical workflow detection is real time operating room (OR) scheduling. For hospitals, each minute of OR time is important in order to reduce cost and increase patient throughput. A logical step in understanding surgical workflow is the real-time detection of what tools are being used and the current phase of the surgery. This is of special concern in laparoscopic surgeries since tool usage data is not readily available. In the medical field researchers have addressed the problem of surgical tool detection for laparoscopic surgeries using video data. Traditional approaches in this field generally tackle the video analysis using hand crafted video features to facilitate the tool detection. Recently, Twinanda et al [1] presented a CNN architecture EndoNet which outperformed previous methods for both surgical tool detection and surgical phase detection.

Given the recent success of these networks, we present a study of various architectures coupled with a submission to the M2CAI Surgical Tool Detection challenge. Upon completion of the challenge a complete paper submission to Arxiv will be made available as an extension to this technical report.

2 Methodology & Experimental Setup

Upon surverying the deep learning literature, we selected three networks that had shown tremendous success in recent competitions. The first of these, Alexnet [2] was one of the first to leverage the benefit of GPUs in training much deeper networks than had been done previously. Following this, various papers expanded on the benefit of neural networks and achieved great success in image recognition challenges. Within these was the VGG architecture [3] which introduced the now commonplace dropout layer to improve performance by combating overfitting. Following this, Szegedy et. al [4] made great strides in efficiently increasing the depth and complexity of the networks with their framework, Inception v3. In this section we will briefly overview the networks, discuss the relevancy of the data they were originally trained on, and the approaches we took to test our

** equal contribution

performance on the provided training data. Prior to discussing each specific approach it is important to note that we took each multi-class image (i.e. an image with a grasper and a specimen bag) and used it for training each of the respective classes in that image. This was done consistently across all methods and is therefore noted here.

2.1 AlexNet Architecture

AlexNet was originally proposed by Krizhevsky et al [2] for the ImageNet object classification task. The architecture consists of an input layer, five convolutional layers and three fully connected layers. For our implementation, we use a pre-trained model and fine tune it using images from the provided training set. The main benefit of the network being pre-trained on the ImageNet dataset from 2012 is the diversity of the classes. The network is able to use similar weights for a much less complex (class-wise) task, lowering the output from 1000 classes to a mere 7 classes. However, it is important to note that the input images are visibly distinct from anything in the ImageNet network. A majority of the image is black because the image is being viewed through an endoscope. The actual content of the image is generally tissue, with the tool being the contrasting object from the background. This poses a trickier problem given that the majority of the images contain a repetitive background with some type of surgical tool which tends to bear the same color and relative shape. There is a lot of room for improving the input data to account for these insights which we discuss in our future work.

Implementation The code was implemented using the Caffe deep learning framework [5]. The fine tuning was done for 50,000 iterations with a learning rate of 0.0001. For evaluation, we train on images from nine videos from the data set and test on the remaining video.

2.2 VGG Architecture

Recently, Simonyan et. al [3] proposed a CNN architecture for object classification which outperformed all previous state-of-the-art architectures on ImageNet classification task at the time. They showed that deeper networks improves the classification accuracy. Therefore, we fine tune their 16-layer CNN model pre-trained on ImageNet for our surgical tool detection problem. This architecture consists of thirteen convolutional layers with 3x3 convolutional filters and three fully connected layers.

Implementation The model was again implemented using Caffe and the fine tuning was done for 70K iterations with a learning rate of 0.0005. Similar to AlexNet implementation, we trained the model on images from nine videos and tested on the last one.

2.3 Inception v3 Architecture

The Inception v3 network (from 2015-12-05) introduces a series of mixed layers to its architecture that have shown a significant performance increases in the ILSVRC 2012 and ImageNet challenges. One of the key contributions of this approach was to combat the representational loss introduced by pooling. The full details of the network and its architecture can be seen in their paper [4]. Inception v3 is similarly trained on the ImageNet data from 2012 which contains 1000 relatively generic object categories. The benefit and detriment of these categories is previously discussed in the Alexnet section.

Learning Rate The main parameter we tweaked for testing this network was our learning rate. It quickly became evident that a significantly large learning rate (0.01) was needed to adapt the ImageNet-trained architecture to the seven classes in the surgical challenge. We tested a variety of much smaller learning rates but saw that performance was not optimal over a constant number of training steps. An alternate approach would have been to increase the number of training steps given that you are lowering the learning rate but we avoided this to prevent overfitting given that some classes had a relatively small number of training examples. We contrast this with our approaches for Alexnet and VGG in which we take the opposing angle of a smaller learning rate for much higher iterations due to a less complex network architecture.

Input Distortions In addition to this, we tested various input distortions in order to combat overfitting. Specifically, we experimented with randomly scaling, cropping, flipping the image, and adjusting the brightness. The first three approaches showed accuracy improvements but brightness adjustments did not have a significant effect. We randomly scaled and cropped the image 15.0% of its original size (determined experimentally), along with randomly flipping the image left to right.

We also considered rotating the image in order to further expand our training data but decided not to as it changes the orientation at which the surgeon is conducting the surgery which has potentially detrimental effects.

Data Handling For this network we trained on 80% of the input data, used 10% as validation and 10% for testing. We considered labeling the provided test data manually in order to use more of the available data for training but given that it was released a week before the deadline we opted for a more reliable training approach in which we could adequately track our progress. Lowering our validation or testing from 10% was not a viable method given the distribution of images available per class (for instance, scissors and bipolar are seldomly used whereas the hook and grasper are consistently in use). It is important to note that we train using only the image at the given frame which has the classification (and therefore not the 25 images which precede the classification).

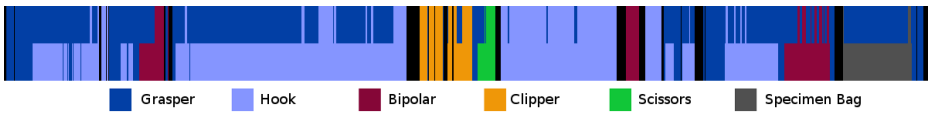

Fig. 1. Visualization of one of the input videos – black is denoted as no tool usage (figure is best seen in color). Each pixel column represents a single frame classification, which is adequately split in chunks if there were multiple classes in a given frame.

Implementation We ran this network on Tensorflow [6] for 4000 iterations with a learning rate of 0.0127. We tested a variety of learning rates and distortions as discussed earlier, and found that 15% randomly introduced scale and crop distortions gave us the best results which we will discuss below.

3 Dataset Discussion

In this section we will briefly discuss a visualization of the input data that gives the reader a better idea of the composition of this multiclass problem in order to better understand the results that we provide. In the example seen in figure 1, we can see that the surgery begins with the grasper and hook, a brief use of the bipolar, followed by the clipper and scissors which are used mid-way through the surgery, and then an intermittent use of the bipolar tool, finishing with the specimen bag. This workflow was characteristic of every surgery provided in that the grasper and hook were used throughout with an intermittent use of the other tools and finishing with the specimen bag.

An interesting correlation in this data is the use of multiple tools at a given time, particularly the use of both the grasper and hook throughout the surgery.

4 Results

Table 1. Comparison of attempted deep learning architectures

Architecture	Accuracy
AlexNet	63.78%
VGG	69.75%
Inception	76.6%

As expected, our best performance was demonstrated from fine-tuning the most recent network architecture, Inception v3, which achieved a 76.6% accuracy. This outperformed both the AlexNet and VGG approaches which achieved an accuracy of 63.78% and 69.75% respectively as shown in Table 1. It is important to note that all of our reported testing accuracies were done on a per-class basis.

We are keen to point out that these comparisons are not entirely fair as the Alexnet and VGG architecture were tested by leaving one of the 10 videos out from testing, whereas the Inception architecture was tested by randomly sectioning a percentage of the input data to be testing and validation.

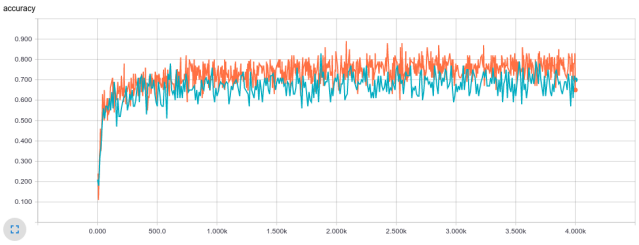

Fig. 2. Training (orange) and validation accuracy (blue) for a fine-tuned Inception network (best seen in color).

We present the training and validation accuracies for the methods in Figure 2, which demonstrate the expected decrease in a change in accuracy. By 4000 iterations, we see the accuracy has stabilized so further iterations are likely to have little effect on the network accuracy with an additional risk of overfitting.

We tested the Inception network on the original videos the network was trained on and obtained an average mAP of 30.61%. We submitted this framework to the M2CAI competition achieving a top-3 mAP of 37.6%.

5 Future Work & Conclusion

In this report, we presented a comparison of different deep network architectures for surgical tool detection and showed that the Inception architecture achieved an accuracy of 76.6% (mAP of 37.6% on the test data).

Firstly, we think that removing the red tissue regions in the images by a simple threshold could help improve the accuracy. Since a high percentage of each image contains flesh but that is generally irrelevant to the task, removing some or all of it would reduce noise and potentially help the network learn better features for tool detection. Along with this, cropping the images to remove the black area from the images could also help improve the classification rate as the images are then rescaled for the network, which currently results in a loss of valuable information. We believe this is why we see an improvement of accuracy when using random crop for our inception approach.

Moreover, incorporating temporal information from the videos can be key to eliminating some misclassifications. For example, the specimen bag is generally used towards the end of the surgery as shown in Figure 1. Such information along with features extracted from the deep network would potentially improve on the accuracy achieved for tool detection.

Lastly, an interesting correlation in the data is the use of multiple tools at a given time, specifically that some tend to be used with others frequently. For example, the grasper and hook shown in Figure 1 tend to be used in unison throguhout large chunks of the surgery. This intuition could be incorporated into the classification in order to continually improve the results.

References

1. Twinanda, A.P., Shehata, S., Mutter, D., Marescaux, J., de Mathelin, M., Padoy, N.: Endonet: A deep architecture for recognition tasks on laparoscopic videos. arXiv preprint arXiv:1602.03012 (2016)
2. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. (2012) 1097–1105
3. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
4. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. arXiv preprint arXiv:1512.00567 (2015)
5. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. arXiv preprint arXiv:1408.5093 (2014)
6. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X.: TensorFlow: Large-scale machine learning on heterogeneous systems (2015) Software available from tensorflow.org.