Identification of Surgical Tools using Deep Neural Networks

Soumali Roychowdhury, Zhengbing Bian, Arash Vahdat, and William G. Macready
{*sroychowdhury, zbian, avahdat, wgm*}*@dwavesys.com*
D-Wave Systems Inc.

I. Introduction

In recent year deep learning has proven very successful on image classification tasks [1], [4], [5], as it enables training massive end-to-end integrated solutions, from a grid of pixels to output classes. The present research work applies deep neural networks to surgical tool annotation, a problem which has been popular to the medical image processing community for the past decade. The goal is not to precisely locate tools in images, but to annotate frames in a video so that an understanding of the surgical workflow can be developed. Our motivation in participating in this challenge is to assist in the first step towards an automated understanding of the most common surgical procedure in the world. The accurate annotation of tools within the video sequence is a necessary first step in building a real-time understanding of the surgical workflow.

II. Experimental Evaluation

The dataset for this challenge consists of 50 videos of cataract surgeries performed in Brest University Hospital. Each frame of these videos is 1920 x 1080 pixels and the frame rate is approximately 30 frames per second. A surgical tool is considered to be in use whenever it is in contact with the eyeball. The supervised annotations of these tools within the video frames was provided by two experts. The cataracts dataset was then divided into training and test sets of 25 videos each.

A. Train and Validation Split

To train our model and evaluate its performance, we further divide the training set into two subparts: a train set and a validation set. The validation set consists of the videos numbered train04, train12 and train21, while the remaining videos are placed in the train set. To support the allocation of videos to the train and validation sets, a histogram of the tool annotations is constructed. We then assign videos to the train and validation sets so that that both sets have the same distribution of tools.

B. Frame Extraction

Instead of extracting training frames at a uniform rate, we densely sample parts of training videos that contain the rare tools (e.g. biomarker, Vannas scissors, etc.). For more common tools, we use the constant 6 fps frame rate. For negative instances, we randomly select 40K frames uniformly from amongst training frames that have no tools. This process provides about 100K training images. All frames are extracted using ffmpeg.

TABLE I
Prediction accuracy of different baselines measured on the validation set using AUC (%).

Network	Input Size	AUC (%)
ResNet-50	540x960	96.78
Inception-v4	540x960	97.86
NASNet-A	270x480	98.64
Aggregated Model	-	99.06
Median Filtering	-	99.41
MRF Smoothing	-	99.66

C. Frame Classifiers

The TensorFlow framework is used for training our models. In earlier submissions, a 50-layer Residual Network, pre-trained on ImageNet, was trained to predict tools in frames. In addition to ResNet50 [3], we train two networks including Inception-v4 [6] and NASNet-A large [7]. Given the GPU memory and run time limitation of these networks we trained them on different image sizes. In Table I, the input size and performance of all the networks on our validation set are reported. For training, a batch size of 4 is used and input images are horizontally flipped and resized with random cropping. All networks are trained using the sigmoid cross-entropy loss function with the Adam optimizer at most for 13 epochs. The learning rate for each network is chosen using cross validation on the validation set using the area under the ROC curve (AUC) measure.

D. Frame-level Post-processing

After training the frame-level networks, their prediction probability scores are aggregated using a weighted geometric mean. The weights are set on the validation set using a grid search. The performance of the aggregated model is shown in Table I.

E. Temporal Smoothing

Temporal smoothing plays a key role in obtaining a smooth, and typically more accurate, prediction for each tool. In previous submissions, we used median filtering for this purpose. In our recent submission, a Markov Random Field (MRF) model is developed to define the probability distribution in the label space. Let's assume that $\boldsymbol{y} = \{y_1, y_2, \ldots, y_T\}$ represents

the binary label vector for a tool where y_t indicates whether or not the tool is present in the t^{th} frame. The MRF model has a chain-like structure and defines a conditional probability distribution for the label vector given the video $\boldsymbol{x}$ using an energy function:

$$p(\boldsymbol{y}|\boldsymbol{x}) = \frac{\exp(E(\boldsymbol{y};\boldsymbol{x}))}{Z}. \quad (1)$$

Z is the normalization constant and

$$E(\boldsymbol{y};\boldsymbol{x}) = \sum_{t=1}^{T} a(s_t)y_t + \frac{1}{2}w\sum_{t=1}^{T}\sum_{n\in N(t)} y_t y_n \quad (2)$$

is the energy function. $N(t) = \{t-19, t-17, \ldots, t-1, t+1, \ldots, t+17, t+19\}$ represents the set of neighboring nodes for the t^{th} frame, and provides long-range temporal connectivity. The total variation in lag of 38 frames corresponds to about 1 second of video. Note that the neighborhood of each node is defined such that the resulting graph is bipartite.

In Eq. (2), $a(s_t)$ is the bias for the t^{th} frame's label which is computed by shifting and scaling the output of the aggregated frame-level prediction score (s_t) at frame t. The coupling parameter w in Eq. 2 enforces label agreement between neighboring frames. The shift and scale parameters for $a(.)$, and w are set on the validation set by a grid search that aims at maximizing the AUC criterion on the validation dataset. Our validation set contains only 14 out of 21 categories, so instead of learning a category-specific shift, scaling, and coupling parameters, we shared these parameters across all 21 categories. Nevertheless, due to the shifting and scaling of the aggregate neural network output (which differs across categories) we form a tool-specific MRF model which is robust against overfitting.

Inference: The MRF model in Eq. (1) represents the joint probability distribution for all the labels in the temporal domain for a tool (category). Given this model, we compute the marginal distribution $p(y_t = 1|\boldsymbol{x})$ using a mean-field approximation [2] and use the resultant marginal probability as the prediction score for the t^{th} frame. Lastly, in order to process videos efficiently we form the MRF model in smaller segments of the length $\sim$20,000 frames instead of the full length of the video. The performance of our MRF model on the validation set is compared against median filtering in Table I.

III. Changes Since the Nov 20th Submission

We train an additional NASNet-A model using a slightly larger image size and by optimizing the weighted sigmoid cross entropy loss. The weights for the sigmoid cross entropy loss are set such that it assigns more importance to the rare categories. This model in conjunction with the previous three models is used for forming the aggregated model. The updated list of models and their performance on the validation set is reported in Table II. Introducing the new network improves our final MRF-based model by 0.11% on the validation set.

TABLE II
Prediction accuracy of different baselines measured on the validation set using AUC (%).

Network	Input Size	AUC (%)
ResNet-50	540x960	96.78
Inception-v4	540x960	97.86
NASNet-A	270x480	98.64
NASNet-A	337x600	98.74
Aggregated Model	-	99.19
Median Filtering	-	99.40
MRF Smoothing	-	99.77

References

[1] Yoshua Bengio. Learning deep architectures for ai. *Found. Trends Mach. Learn.*, 2(1):1–127, January 2009.

[2] Michael I. Jordan, Zoubin Ghahramani, Tommi S. Jaakkola, and Lawrence K. Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233, Nov 1999.

[3] He Kaiming, Zhang Xiangyu, Ren Shaoqing, , and Sun Jian. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[4] Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. 2009.

[5] Yann LeCun. Deep learning. *Nature*, 521, 2015.

[6] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, pages 4278–4284, 2017.

[7] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. *arXiv preprint arXiv:1707.07012*, 2017.