Multi-label classification of surgical tools with convolutional neural networks

Jonas Prellberg
jonas.prellberg@uni-oldenburg.de

***Abstract*—A 50-layer Residual Network is pretrained on ImageNet and finetuned for the CATARACTS challenge. A weighted loss function is employed to improve generalization.**

I. Challenge

This document is part of a submission to the CATARACTS challenge: https://cataracts.grand-challenge.org/home/

II. Algorithm overview

The system called RToolNet processes single video frames using a convolutional neural network and predicts scores for each surgical instrument.

The network architecture is shown in table I. It is based on the 50-layer Residual Network [1] which was pretrained on ImageNet. The weights of the first 31 convolutional layers are frozen so that they are not changed during a subsequent training step. The output of the last (49th) convolutional layer is fed to a global average-pooling followed by a fully connected layer with 21 units and a sigmoid activation function.

Each of the 21 network outputs corresponds to a predicted score for a surgical instrument. Because the instruments are not mutually exclusive, the task is treated as 21 separate binary classification problems. These are trained using a cross-entropy loss function. Before the backwards pass, the cross-entropy loss associated with each output (class) i is multiplied by the class weight

$$w_i = \sqrt{\frac{\max\{f_j \mid 1 \leq j \leq 21\}}{f_i}}$$

where f_i is the frequency with that class i appears in the dataset. All 21 output losses are averaged to get the total network loss.

The network is trained using single frames extracted from the training videos. Due to the very large amount of images that don't show any instruments, 60 % of those images are randomly discarded. Additionally the videos are sampled at 5 frames per second because subsequent frames are very similar to each other. Each image is scaled down to 960×540 pixels and normalized by subtracting mean pixel activations. To increase the variance in the training images the following augmentations are performed randomly: cropping, rotation, horizontal flipping, and color augmentation as described in [2].

The training procedure is stochastic gradient descent with a learning rate of 0.05 and a momentum term of 0.9. The learning rate is decayed over time. For each batch n of 8 frames the learning rate is calculated as $0.05 \,/\, (1 + 0.000125n)$ and a total of 25k batches are processed for the training. During inference about 26 images per second can be processed using a NVIDIA Tesla P100.

Table I
Network architecture

Layer	Configuration	Output size	
Input	540×960	540×960	3
ResNet50	c. layers 0 to 31 frozen output of c. layer 49	17×30	2048
GlobalAvgPool		—	2048
FC	21 units sigmoid activation	—	21

III. Relation to previous submissions

Instead of using ResNet as a fixed feature extractor paired with a custom architecture, ResNet is now completely reused and finetuned. This requires more time and memory for training but also improves the network's performance. Furthermore, a weighted loss function is used to improve generalization slightly. Finally, the data pipeline that feeds the network is now faster which benefits both training and inference times.

References

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems*, NIPS'12, pages 1097–1105, USA, 2012. Curran Associates Inc.