DenseNet for cataract surgical tools classification

Fenqiang Zhao, Chenghui Qiu, Shunren Xia
Zhejiang University
zhaofenqiang@zju.edu.cn, qch19881114@126.com, srxia@zju.edu.cn

Abstarct

This article is a draft for cataract surgery tool annotation challenge, also known as CATARACTS Challenge [1]. In this paper, we proposed a convolutional neural network for surgical tools classification based on DenseNet169 [2], and we refer to it as Cataract DenseNet (CDenseNet).

1. Introduction

Convolutional neural network (CNN) have become the dominant machine learning approach for visual object recognition. As the computer hardware become more and more powerful, we can train very deep CNN with properly composed structure. DenseNet [2] introduces direct connections between any two layers with the same feature-map size. Benefiting from improved flow of information and gradients throughout the network, DenseNet is very easy to train and scaled to hundreds of layers, leading to state-of-art performance over many computer vision tasks.

This challenge aims to indicate which tools are being used by the surgeon at each instant in the cataract surgery. The dataset consists of 50 videos (25 for train and 25 for test) of cataract surgeries, in which the presence of 21 surgical tools was manually annotated by two experts, and a tool was considered to be in use whenever it was in contact with the eyeball.

2. Algorithm Overview

The CNN we proposed called CDenseNet is based on DenseNet169 [2], and the last fully connected layer consists of 21 units for predicting the probability of the corresponding tool existence respectively, and the average pooling layer's size was modified to fit the cataract surgery image's size. We adopted 4 dense blocks with {6, 12, 32, 32} convolution units respectively, and the growth rate for this networks is 24, the number of initial convolution layer's feature maps is 64. The exact network configurations are shown in Table 1.

At first we extracted video frames every 5 frames, and resized the image to 540×960 into train dataset, and 1/3 of the train dataset samples were selected into valid dataset. To overcome the imbalance of the datasets, we purposely extract more images from video14, video19, video24 and video25, for mendez ring, needle holder, biomarker, cotton and vannas scissors, which all have few images in the train datasets. We also converted all "0" in ground truth to "-1", for adopting weighted binary softmargin loss function, the weight is inversely proportional to the number of corresponding tool's samples, and 1 for none of the tools existence sample:

Layers	CDenseNet	Output Size
Convolution	7 × 7, stride 2	270 × 480
Pooling	3 × 3 max pool, stride 2	135 × 240
Dense Block (1)	$\begin{Bmatrix}1\times 1\text{ conv}\\3\times 3\text{ conv}\end{Bmatrix}\times 6$	135 × 240
Transition Layer (1)	1 × 1 conv	135 × 240
	2 × 2 avg pool, stride 2	67 × 120
Dense Block (2)	$\begin{Bmatrix}1\times 1\text{ conv}\\3\times 3\text{ conv}\end{Bmatrix}\times 12$	67 × 120
Transition Layer (2)	1 × 1 conv	67 × 120
	2 × 2 avg pool, stride 2	33 × 60
Dense Block (3)	$\begin{Bmatrix}1\times 1\text{ conv}\\3\times 3\text{ conv}\end{Bmatrix}\times 32$	33 × 60
Transition Layer (3)	1 × 1 conv	33 × 60
	2 × 2 avg pool, stride 2	16 × 30
Dense Block (4)	$\begin{Bmatrix}1\times 1\text{ conv}\\3\times 3\text{ conv}\end{Bmatrix}\times 32$	16 × 30
Classification Layer	16 × 30 avg pool	1 × 1
	21D fully-connected	21

Table 1: CDenseNet architecture

$$loss(x,y) = \sum_{i=1}^{N} w_i \log(1 + e^{-y_i x_i}) \quad (1)$$

In formula (1), N is 21, indicating the 21 independent binary classification problems, y is the ground truth label only containing either "1" or "-1", representing the positive and negative sample respectively, and x is the output of the fully-connected layer without softmax activation.

To train the network, we use stochastic gradient des-cent (SGD) with a weight decay of 0.0001 and a momentum of 0.9 without dampening. The initial learning rate is set to 0.05, and is divided by 10 at 30% and 60% of the total number of training epochs. The weight is initialized by gaussian function. Due to GPU memory constraints, we train the network with a mini-batch size 8 for 15 epochs. During training, images are firstly normalized by subtracting mean and dividing by standard deviation, and then we also adopted standard data augmentation scheme, such as random horizontal flip and random crop with padding 40 blank pixels.

After training, we evaluate our approach on the valid dataset using the given script by computing the area under ROC curve. By this way, we made some parameter optimization and furthermore cross validation will be done to get better performance.

It is worth noting that our CDenseNet is actually efficient in parameter utilization and memory consumption following the implementation in [3]. We believe that further gains in accuracy of CDenseNet may be obtained by deeply structured CDenseNet architecture and detailed tuning of hyperparameters.

Finally, about 24 images per second can be processed during inference using a NVIDIA 1080 Ti.

References

[1] CATARACTS. 2017. Challenge on Automatic Tool Annotation for cataRACT Surgery. https://cataracts.grand-challenge.org. (2017).

[2] Huang G, Liu Z, Weinberger K Q, et al. Densely Connected Convolutional Networks[J]. 2016.

[3] Pleiss G, Chen D, Huang G, et al. Memory-Efficient Implementation of DenseNets[J]. 2017.