CS760: Project Report

Haiyun Jin
University of Wisconsin-Madison
hjin38@wisc.edu

Qinheping Hu
University of Wisconsin-Madison
qhu28@wisc.edu

Yuzhen Wang
University of Wisconsin-Madison
ywang855@wisc.edu

Lu Li
University of Wisconsin-Madison
lli375@wisc.edu

Xiuyuan He
University of Wisconsin-Madison
xhe75@wisc.edu

Abstract

With a widespread use of digital imaging data in hospitals, the size of medical image repositories is increasing rapidly.

Keywords Cervix cancer, Convolutional neural network, GENIC

1. Introduction

1.1 Introduction to Cervical Cancer

Cervical cancer is a cancer arising from cervix due to abnormal growth of cells that have the ability to invade or spread to other parts of the body. With no typical symptoms be seen early on, there may be abnormal vginal bleeding, pelvic pain and so on later symptoms. Most of cervical cancer cases are caused by Human Papillomavirus(HPV). Survey shows that every year in the United States, HPV causes 12,900 new cases of cervical cancer, which is about 35 women diagnosed each day. Women in their teens and 20s are more vulnerable to certain infections than older women. Also, smoking, both active and passive, is a risk factor for cervical cancer besides HPV. It can directly induce cervical cancer by producing CIN3 with higher probability which has the potential of forming cervical cancer.

1.2 Prevention and Diagnosis

There are several methods for prevention. Screening by Papanicolaou test or Pap smear every 3-5 years will help reduce number of cases of cervical cancer. Liquid-based cytology is intended to improve the accuracy of the Pap test. Barrier protection like condoms and vaccination method like HPV vaccines can reduce the risk of infection significantly. As for diagnosis, biopsy is a crucial method, which is often done through colposcopy.

1.3 Convolutional Neural Networks

Convolutional neural network(CNN, or Convnet) is a type of feed-forward artificial neural network in which the connectivity pattern between its neurons is inspired by the organization of the animal visual cortex. Convolutional networks were inspired by biological processes and are variations of multipayer perceptrons designed to use minimal amounts of preprocessing.

1.4 Convolutional Neural Network's Role in Medical Image Processing

As a type of deep learning neural networks, convolutional neural networks have been used in image classification, semantic segmentation and many other fields. For example, segmentation need to delineate different anatomical structures and detect unhealthy tissue in human body.

2. Methodology

2.1 Dataset Description

Intel & MobileODT Cervical Cancer Screening Kaggle Competition is developed by Intel partnered with MobileODT. In this competition, we will develop an algorithm to accurately identify a woman's cervix types based on images. The goal of this is to prevent ineffectual treatment and allow healthcare provide to give proper referral for cases that required further treatment. The data set we used consists of 8203 images, classified in 3 categories. The labels are not evenly distributed with a skew on 'Type_2'. The distribution of three types are 17.51%, 52.94%, 29.55%, respectively. We notice that a random player will have 33% chance to predict a correct type, significantly lower than 2-label classification problem. We also notice that if a model predicts all 'Type_2', it will get about 50% chance correct. Thus our goal is to have an accuracy better than 50%, not a random player.

2.2 Data preprocessing

2.2.1 Downsample the original images

Most of the original images are more than 4000x3000 pixels in size. These are too much information redundancies and also a nightmare for model training. As we recognized that we can downsample the images to lower resolution without losing important features. In fact, in the image classification area, most models are developed based on images that have only hundreds of pixels per dimension. We started to downsample the image to 640x480, 320x240, 160x120, 100x100, 64x64, and eventually 32x32. The larger size input image does not increase the accuracies significantly thus we choose 32x32 size with a balance of the accuracy and efficiency for the model training.

2.2.2 Image Segmentation

2.2.3 Data augmentation

The whole training set size is 8203. We decided to save 821 examples as the test set that will never be touched during the training process. Although the rest of 7382 samples seems to be a large training set, it is still much too small compared to the number of parameters a CNN usually has. Thus we used ImageDataGenerator[Keras website] method in Keras framework to zoom, rotate, shift, or combination of all above to artificially increase the size of training set. For each model, the size of training set is augmented to 13288.

2.3 Machining Learning method

2.3.1 k-nearest neighbor(k-NN)

The first approach we pursued is k-nearest neighbor. The 32x32x3 images are reshaped to 1x3072 and classified results are shown in the following table. Surprisingly, the 1-NN

Table 1: k-NN prediction accuracy

k	Accuracy%
1	66.26
3	61.75
5	59.20
10	59.20
15	57.37

models is the best model among the several model we chose. And the there is trend that the accuracy on test set decreases with more neighbor is referenced. 1-NN gives 66% accuracy which is a reasonable good model.

2.3.2 Multi-layer perceptron

The next model we tried is multi-layer perceptron. The model is described below. We evaluate the model by 5-fold cross validation the results is shown in the following table. As we see, the test accuracies on all 5 folds are not even comparable with 1-NN classifier. However, if we take the average confidences of the output unit to predict the types, the accuracy is improved over any individual model and comparable to 1-NN classifier.

Table 2: MLP model

Dense(512), activation='sigmoid'
Dropout(0.1)
Dense(512), activation='relu'
Dense(3, activation='softmax')

Table 3: Cross validation on MLP model

Folds	valid Acc.%	Test Acc.%
Fold 1	62.69	64.07
Fold 2	62.15	62.85
Fold 3	58.50	58.10
Fold 4	64.09	62.24
Fold 5	63.25	60.54
Average	62.14	
Avg weight		65.04

2.3.3 Convolutional Neural Network

Convolutional Neural Network is powerful model in image processing. However, there is no universal architecture that we can naively use. We start from simple architecture and gradually increase number of layers or convolutional filters.

With increasing number of convolutional filters and hidden nodes, the prediction accuracies on test set gradually increase from worse than 1-NN classifier to over 70%. The first two ConvNet are trained for 100 epochs and the later two are trained for only 50 epochs because of the training inefficiency. However, we are still confident in that the epochs are large enough to get a converged results since the training accuracies are close the 95% but the validation accuracies are around 65%. This is an indication of the start of overfitting thus we can safely stopped the training.

As shown in Table 5, The 'intermediate size' model has the best test accuracy of over 71

2.3.4 Ensemble/Random forest

With all the models we build, we create a model ensemble of all the models and take the majority vote as the output. The results improve slightly over all individual model

2.3.5 Bagging/Boosting

As discussed in the Sec. Method, data augment increases the size of the train set but the model does not show improved the test accuracy.

3. Implementation

The k-NN model is build on sci-kit learn package and the MLP, CNNs are build with Keras with Tensorflow as backend.

Table 4: CNN model details

Conv1	Conv2	Conv3	Conv4
Conv2D(16, (3, 3), relu)	Conv2D(32, (3, 3), relu)	Conv2D(64, (3, 3), relu)	Conv2D(64, (3, 3), relu)
MaxPooling2D((2, 2))	MaxPooling2D((2, 2))	MaxPooling2D((2, 2))	Conv2D(256, (3, 3), relu)
Conv2D(32, (3, 3), relu)	Conv2D(64, (3, 3), relu)	Conv2D(128, (3, 3), relu)	MaxPooling2D((2, 2))
MaxPooling2D((2, 2))	MaxPooling2D((2, 2))	MaxPooling2D((2, 2))	Conv2D(512, (3, 3), relu)
Dropout(0.2)	Dropout(0.2)	Dropout(0.2)	MaxPooling2D((2, 2))
Flatten()	Flatten()	Flatten()	Dropout(0.2)
Dense(64, tanh)	Dense(128, relu)	Dense(256, relu)	Flatten()
Dropout(0.1)	Dropout(0.1)	Dropout(0.1)	Dense(1024, relu)
Dense(3, softmax)	Dense(3, softmax)	Dense(3, softmax)	Dense(256, relu)
			Dropout(0.2)
			Dense(3, softmax)

Table 5: CNN results

	conv1		conv2		conv3	
	Vaild Acc.	Test Acc.	Vaild Acc.	Test Acc.	Vaild Acc.	Test Acc.
Fold 1	57.35	57.00	63.37	68.09	65.67	64.80
Fold 2	56.94	56.03	66.08	67.48	66.89	68.57
Fold 3	58.56	58.10	65.67	64.92	65.54	66.75
Fold 4	58.20	57.13	65.31	67.48	66.67	65.77
Fold 5	59.05	57.61	66.10	67.48	65.97	68.09
Average	58.02		65.31		66.15	
Avg Weight		62.73		71.62		71.01

Table 6: Ensemble Models

Model	Accuracy
conv1	62.73
conv2	71.62
conv3	71.01
conv4	65.77
mlp	65.04
1-nn	66.26
ensemble	71.51

4. Evaluation

5. Related work

6. Conclusion

Acknowledgements

7. Future Work

Transfer learning is a process that initialized the weights of a model to what have been learned from a previous dataset. Our future includes a fine-tuning of the VGG 16, a very deep neural network pre-trained on ImageNet dataset. The VGG 16 weights have more than 138 million parameters and the weight file itself is as large as 588 MB. Fine-tuning of such a large parameters space requires smaller learning rate thus again slows down the training process.

References