Identifying Cervix Types using Deep Convolutional Networks

Lei Lei
Stanford University
leilei@stanford.edu

Ruoxuan Xiong
Stanford University
rxiong@stanford.edu

Huaiyang Zhong
Stanford University
hzhong34@stanford.edu

Abstract

We implemented and fine tuned four different architectures of convolutional neural networks on the problem of identifying cervix type. We experimented with multiple choices of hyper parameters, transfer learning from ImageNet, as well as manually identifying region of interest to improve classification accuracy. Experiments showed Inception v3 outperforms other architectures, such as AlexNet, ResNet and VGG16. Training the model from scratch outperformed fine-tuning pre-trained models, and identifying region of interests helped classification of one type of cervix. Saliency maps show that Inception v3 exhibited desirable neural activation in the region of interest. A bounding box regressor exhibited good performance and indicated the possibility of adapting R-CNN in the future work.

1. Introduction

Cervix cancer is one of the most common cause of cancer death for women. However, cervix cancer is easy to prevent if women can get effective treatment at an early stage. Every woman should have access to it. The treatment can be very different depending on patients physiological differences. Wrong treatments are costly and can incur high health risks. Developing an appropriate method of treatment for individual woman is important but not easy in the areas lack of expertise in this field, especially in the rural parts of the world.

There are three types of cervices. If the cervix type of a patient were correctly identified, appropriate treatment can be provided. Intel & MobileODT Cervical Cancer Screening is an image classification challenge on Kaggle aimed at differentiating three physiologically different types of cervix. If a good algorithm can be developed, it can be integrated into the digital toolkit for health care workers of every level to provide expert services to patients. In this paper, we will use cervical images as inputs, feed them into Deep Convoluntional Neural Network (DCNN) with various architectures, and predict the types of cervices.

2. Related Work

DCNN have achieved a great success in computer vision, especially in the field of image classification (e.g. Ciresan et al.[3],Ciregan et al.[4], Krizhevsky et al.[10],Simonyan et al.[17]). Because of rapid development of computer vision thanks to the intensive research in deep learning in recent years, more and more professionals and experts from other fields start trying transferring the success of image classification with DCNN to their fields. Medical image classification is one of those fields that have drawn special attention partially due to the extraordinarily large amount of data generated from electronic medical record (EMR) and the need to understand the data[15]. The publications (e.g. Ronneberger et al.[14], Ciresan et al.[5], Milletari et al.[12], Kamnitsas et al.[9], Havaei et al.[6]) in the field of medical image classification/segmentation have become one of the most popular application areas of DCNN and medical societies have become more open to the aid of machine learning. However, until now, none of publications have examined the ways of classifying cervix type probably to due to the lack of data. The current standard procedure of identifying the cervix type depends purely on the judgment of the gynecologist which is prone to human errors and have high costs. Such approach is not sufficient and economically feasible in a resource-limited settings such as Sub-Saharan Africa and South Africa where have higher the prevalences of cervical cancer but have limited amount of medical resources[1]. The aid of a well validated classification algorithm to identify the cervix type prior is of vital importance in helping prevent cervical cancer and improve the treatment of cervical cancer.

3. Methods

We survey four well-known deep convolutional neural network architectures, AlexNet (see [11]), VGG16 (see [17]), GoogleNet (see [18]) and ResNet (see [8] (see table 1) that have shown good performances on ImageNet. The last few fully connected layers are modified to suit this classification task (see detailed explanation below).

1. **AlexNet:** we use exactly same architecture as [11], ex-

cept model output is the softmax scores for three types of cervices. It has 5 convolutional layers and 3 fully connected layers, with around 12.6 million trainable variables.

2. **Inception v3:** we use exactly same architecture as [11], except the last layer adapted to our output (we referenced the implementation here [1]). This model has 21 convolutional layers and 1 fully connected layer. Inception v3 has fewest number of trainable parameters compared to other architectures.

3. **VGG16** has 13 convolutional layers and 3 fully connected layers (we referenced the implementation here [2]). To fit into the memory limit of a single GPU and speed up training, we use 3×3 filters for each convolutional layer, but reduce the number of filters by half. The output dimensions for last three fully connected layers are 512, 1024 and 3 respectively. We apply dropout with probability 0.8 to the first fully connected layer. The adjusted VGG16 model has comparable number of trainable parameters as AlexNet.

4. **ResNet** we build our model based on the 34-layer architecture in [8] (we referenced the implementation here [3]. It has 33 convolutional layers, with 4 residual blocks. We add 3 fully connected layers on top of the convolutional layers, whose dimensions are 512, 1024 and 3 respectively. We have 2 more fully connected layers compared with 34-layer architecture in [8], which result in better performances. The side effect is that our ResNet has 21.6 million trainable parameters, which is the most among four architectures we choose.

With these minor adjustments, mostly to the fully connected layers, the convolutional layers in the adjusted models can still have the receptive fields covering the whole image. These convolution layers can capture many generic nonlinear features such as shape and angle, that are useful in the cervix classification task. The fully connected layers are adapted to cervix classification task at hand, trying to find linear combination of nonlinear features used to identify cervix types, which may be different from that used to classify images on ImageNet.

4. Dataset and Features

The input to this problem are 5278 training images of size with labels $\in \{1, 2, 3\}$ (see Table 2) and 512 test images

[1] https://github.com/tflearn/tflearn/blob/master/examples/images/googlenet.py

[2] https://github.com/tflearn/tflearn/blob/master/examples/images/vgg_network.py

[3] https://github.com/tflearn/tflearn/blob/master/examples/images/resnext_cifar10.py

...	AlexNet	Inception v3	VGG16	ResNet
Layers	8	22	16	36
Trainable Parameters	12.6M	6.17M	12.6M	21.9M
Training Time Per Epoch (1320 images)	$\sim$17 s	$\sim$ 42 s	$>$ 1 min	$\sim$ 33 s

Table 1: Summary of four architectures used (Training Time is measured by single GPU (Tesla K80) with mini-batch size equal to 64)

Cervix Type	Type 1	Type 2	Type 3	Total
Main	249	781	450	1480
Additional	1187	639	1972	3798

Table 2: A summary of distribution of class labels

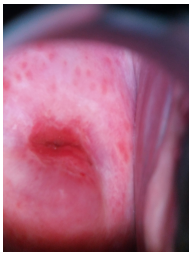
without labels. The training images are partitioned into two sets, main and additional sets. Images from the the main training set are of high image quality (see Fig. 1). Images from the additional training set sometimes come from duplicated patients while sometimes are of low image quality (see 2). We randomly select 90% of images in the main data set as the training set and rest are the validation set. In the attempts to use images in the additional data set to train Inception v3, the model takes longer time to train, is harder to converge, and unfortunately has a higher validation loss compared to the model only trained by images in main set. The remainder of this paper only presents results obtained with main set as training set.

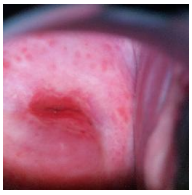
All training images are JPEG file with scale $3 : 4$ and most frequent sizes are 2448×3264 and 3096×4128. We pre-process them down to $256 \times 256 \times 3$ RGB images (see Figure 1) through the following steps:

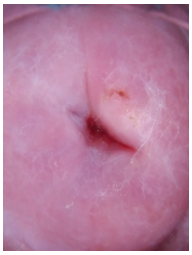
1. Scale all images down to 256×256 by affine transformation on the four corners with bi-cubic interpolation on the interior grid points.

2. Shift all training images by the mean evaluated over all training samples by RGB channels. We center the validation and test samples by the same channel-wise mean.

The input images do not all have high quality (see Figure 2); some are out of focus and some only has a single color channel. In addition, many images have the vaginal speculum, which is inserted into the vagina to dilate it for examination of the vagina and cervix. Some images have a small region of cervix compared with the speculum. In the second step of data pre-processing, we manually identify the Region of Interest (ROI) for all images in the main


original image of Cervix Type 1


preprocessed image of Cervix Type 1


original image of Cervix Type 2


preprocessed image of Cervix Type 2


original image of Cervix Type 3


preprocessed image of Cervix Type 3

Figure 1: Preprocessing: on the left are images from data sets and on the right are preprocessed images by scaling them down to fixed size 256×256 with affine transformation on the four corners and bicubic interpolation on the interior pixels.

Figure 2: Some examples of bad training image

set. We extract ROI and scale all ROIs up to $256 \times 256 \times 3$ to feed into our models.

During the training stage, images are augmented through the following steps:

1. randomly crop
2. randomly flip an image left to right
3. randomly rotation an image by a random angle from -45° to 45°.

A standard color channelwise mean is calculated for all training image and subtracted for each batch as the final preprocessing step.

5. Results

Our objective is to minimize the loss function on the set of test image, defined as a categorical cross entropy function:

$$L = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\log(p_{ij}) \quad (1)$$

where p_{ij} is the probability that observation i is predicted to belong to class j and y_{ij} is the 1 when observation i is belong to class j and 0 otherwise. We add l_2 regularization loss for the weights to 1 in the train process.

5.1. Training Deep Convolutional Neural Network from Scratch

We initialize the weights in the convolutional and fully connected layers from with normal distribution and the bias to zero. We train each model up to 80 epochs. The training will stop early if validation accuracy does not improve in 10 epochs. We tune learning rates, number of fully connected layers and their dimensions and optimizers for each of the four architectures. The best model for each architecture is selected by the best validation loss/accuracy on the validation set. We use the best model to predict class scores of images on the test set and submitted them to kaggle. The loss is evaluated by Kaggle.

The hyper-parameters chosen for each model architecture are summarized in Table 3. We choose a higher dropout probability than proposed in the original AlexNet and VGG16, as the input size is significantly less than number of trainable parameters. We choose RMSprop over SGD with momentum for its faster rate of convergence and stability near local minimum of loss function. We find that using momentum updating rule can easily cross the local minimum and diverge on the cervix dataset. In contrast, RMSprop can generate a roughly monotonically decaying training and validation loss.

The results for four types of models see Table. 4. Inception v3 has best validation accuracy and loss and test loss, followed by ResNet. The train accuracy and loss are very close to validation accuracy and loss for both inception v3 and ResNet. The training loss history and accuracy of Inception v3 and Resnet see Fig. 4. Inception v3 has consistent better performance from the first few epochs

...	AlexNet	Inception v3	VGG16	ResNet
L2 reg.	0.001	0.001	0.001	0.0001
lr	0.001	0.001	0.0001	0.001
Dropout	0.8	0.4	0.8	0.8
Optimizer	Mom.	Mom.	RMSprop	RMSprop
Mini-batch size	64	64	64	64

Table 3: Hyper-parameters choices for 4 Types of Deep Learning Models

...	AlexNet	Inception v3	VGG16	ResNet
Training Acc.	65%	67%	71%	69%
Training Loss	0.79	0.74	0.68	0.79
Validation Acc.	60%	73%	63%	64%
Validation Loss	0.89	0.66	0.91	0.83
Test Loss	0.92	0.79	1.72	0.82

Table 4: Results from 4 Types of Deep Learning Models

compared to ResNet. From the trend of the loss history, the performance of our models could be better if we train more epochs, but the loss and accuracy fluctuate severely, and consequently the loss on the validation set and test set is more sensitive to the checkpoint we choose than to the number of epochs we train after we train 50 epochs for both Inception v3 and ResNet.

It is worth noticing that with 1320 training images, none of the models exhibit overfitting when dropout probability set to zero in our search of best hyperparameters. Our hypothesis is that these architectures may be too complicated for this data set and have too many local minimums and our choices of learning rate update scheme could easily jump over these local minimums.

Table 5 shows the percentage of predicted labels for each type of cervix for Inception v3. Inception v3 has highest accuracy to identify type 2 images, which is 80%. The accuracy to identify type 1 and 3 is around 60%. Even though the accuracy to identify type 3 images is high, it comes with the very high false positive rate, that is, around 40% of type 1 and 3 images are mistakenly identified as type 2 images. A possible explanation for this result is that around 53% images are type 2 cervix. It is possible that if the distribution used to sample each batch size for training has slightly type 1 and 3 probabilities that the probabilities in the empirical distribution calculated from Table 2, we could mitigate the problem that the prediction is biased significantly towards type 2.

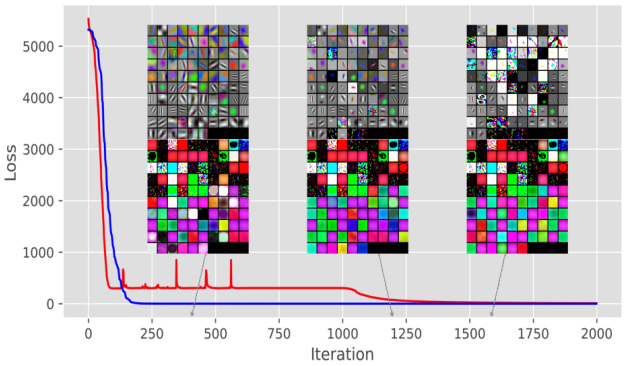


Figure 3: Visualization of conv1 filter in AlexNet. Top row: history of conv1 filter initialized from pretrained Alexnet and fine tuned on the data set. Bottom row: history of conv1 filter initialized from random and fine tuned on the data set.

5.2. Effect of Transfer Learning

5.2.1 AlexNet

We measured the effect of transfer learning on AlexNet with two modes of training AlexNet for 80 epochs on the main dataset. In the first mode, all trainable weights are randomly initialized and trained via stochastic gradient descent (SGD) with learning rate = 10^{-5} and momentum = 0.9. In the second mode, all trainable weights are initialized with weights obtained on ImageNet, and trained via two stages: in the first stage (48 epochs), only weights from the top 5 layers are trained via SGD with learning rate = 10^{-5} and momentum = 0.9. In the second phase(32 epochs), the momentum are set to zero and all weights are trained via SGD with a smaller learning rate =10^{-6} and momentum = 0.9. Both modes demonstrated similar loss history profiles in the training set in Figure 3.

We visualize the effect of transfer learning through the bottom convolution layer in Figure 3. It is clear that conv1 filters loss all its "good" features when fine tuned on the main dataset and do not look much better than trained from scratch. We also noticed that the class scores reported on the test images by the pre-trained AlexNet are predominantly sampled from neighborhood of $[0.08, 0.72, 0.20]$, thus predicting type 2 labels for *all* images. We conducted similar experiment on VGG16 and ResNet and observed the same effects.

5.2.2 VGG16 and ResNet

We performed similar analysis on VGG16 and ResNet with our own custom fully connected layers (see Section 5.1). Due to presence of skip connections, we did not take the

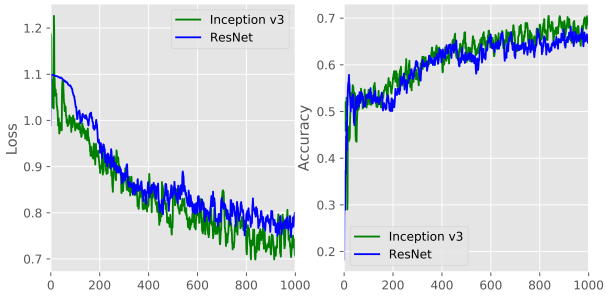


Figure 4: Training loss and accuracy of Inception v3 and ResNet

two-stage fine-tuning approaches for transfer learning; instead, the training is performed on all weights for 80 epochs with parameters specified in Section 5.1. The same two phenomenons are observed: one, VGG16 and ResNet are stuck in the local optimal in fewer than 3 epochs for all the parameter update methods, initial learning rate and l_2 regularization we tried so far and reported class scores $[0, 1, 0]$ for all test images; two, no visually good conv1 filter were obtained in both cases.

5.2.3 Our Hypothesis

One possible explanation is that the cervix images are quite different from the images on ImageNet. On the one hand, we think we need faster learning rate to quickly make adjustment to the differences from ImageNet. On the other hand, we think we need slower learning rate to avoid the fast trap in local optimal. We are still figuring out the best way to make use of pretrained weight. In the future We may use pretrained weights for the bottom convolutional layers, especially to bottom layers, and may initialize bias in the top layers to allow adjustment in our model.

5.3. Saliency Map Visualization

Given that none of DCNN exhibiting overfitting, we plot saliency map to visualize the neural activation for each type of images. A saliency map tells us the degree to which each pixel in the image affects the classification score for that image [16]. To compute it, we compute the gradient of the unnormalized score corresponding to the correct class with respect to the pixels of the image.

Fig. 6 shows the the saliency map for Inception v3. For type 1 images, those with highest classification scores are affected by the red patch most, which is expected; For type 2 images, those with highest classification scores are affected by the black region in the middle of the image, which is expected as well. For type 3 images, those with highest classification scores are affected by the boundary of images and shadow of cervix, which should not be the useful information to identify type 3 cervix. As a consequence, most type 1 and type 2 images with lowest classification scores have virginal speculum and shadow of cervix, because they are "confidently" identified as type 3 cervix by Inception v3.

True\ Pred.	Type 1	Type 2	Type 3
Type 1	0.57	0.40	0.03
Type 2	0.15	0.80	0.05
Type 3	0.01	0.39	0.60

Table 5: The distribution of predicted types for each cervix type in Inception v3

The results from saliency maps motivate us to focus on deactivating neurons that detects virginal speculum and shadow of cervix and is the motivation of manually label Region of Interest (ROI) in our pre-processing steps.

5.4. Effect of Bounding Box

We manually draw and crop the bounding box for the region of cervix in images on the main set. Since region of cervix has different shape and size, we scale the regions up to $256 \times 256 \times 3$ images to feed into the same neural network, so that we can measure the effects of cropping. We name the images only with cervix as cropped and scaled images.

5.4.1 Classification Result and Saliency Visualization

We use Inception v3 to train the scaled images and Table 6. The training loss and accuracy of Inception v3 using original images and cropped and scaled images see Fig. 5. Inception v3 using cropped and scaled images dominates the one using original images from the start of the training process evaluated by either loss or accuracy. In addition, Inception v3 with cropped and scaled images achieved the loss of 0.64 and the accuracy of 76% on the validation, which is better than the results using the original images in Table 4.

Saliency maps 7 show that the region that affects classification score most is larger compared with the saliency maps 6 using the original images, because the region of interest of the cropped and scaled images spread the whole images. The distribution of predicted classes for each cervix type see Table 6. Compared with Table 5, the accuracy for type 3 cervix identification significantly increases. Cropping the irrelevant information out forces the model to learn from the useful information to identify cervix. Type 2 and 3 cervices identification benefit most from cropping.

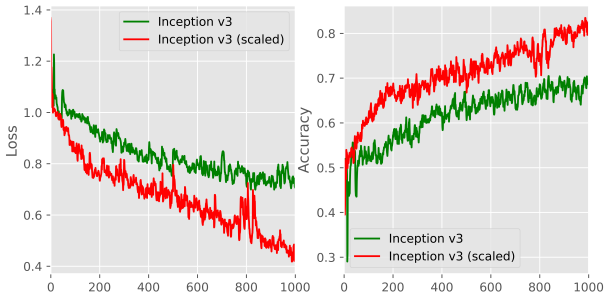
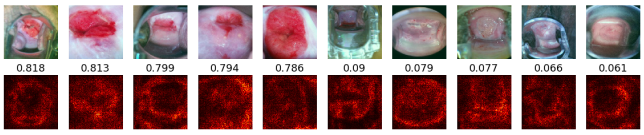
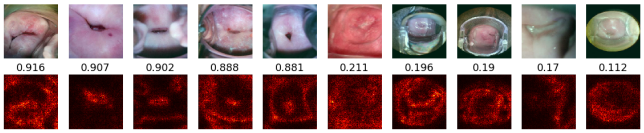


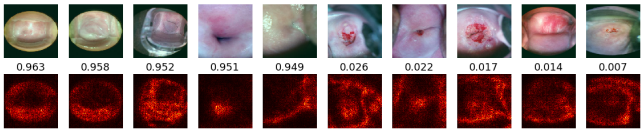
Figure 5: Training loss and accuracy of Inception v3 using original images and cropped and scaled images



(a) Type 1



(b) Type 2

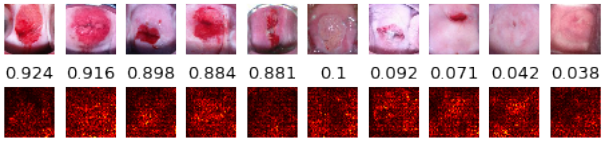


(c) Type 3

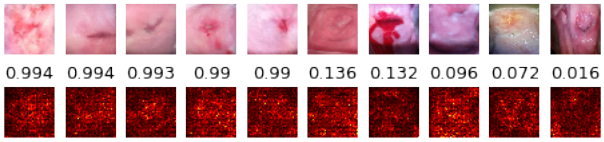
Figure 6: Saliency Map for Inception v3 for the top 5 and bottom 5 classification scores corresponding to the correct class (the classification score is between the original image and the saliency map)

True\ Pred.	Type 1	Type 2	Type 3
Type 1	0.55	0.39	0.06
Type 2	0.09	**0.86**	0.05
Type 3	0.05	0.17	**0.78**

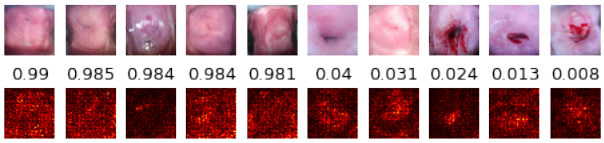
Table 6: The distribution of predicted types for each cervix type in Inception v3 (using cropped images within bounding box for training)



(a) Type 1



(b) Type 2



(c) Type 3

Figure 7: Saliency Map for Inception v3 (using cropped images within bounding box for training) for the top 5 and bottom 5 classification scores corresponding to the correct class (the classification score is between the original image and the saliency map)

5.4.2 Bounding Box Prediction

Since using cropped and scaled images can have lower validation loss and higher accuracy, in this section, we are interested in predicting the bounding box coordinates given images. Faster R-CNN [13] is the state-of-art model to detect object. We build our bounding box prediction model similiar to Faster R-CNN; we do not use Faster R-CNN as we think our problem is not a multi object detection problem. The objective is to predict (x_i, y_i, w_i, h_i), where (x_i, y_i) is the center of the bounding box and w_i and h_i are width and height of the bounding box. Let the predicted bounding box has $(\hat{x}_i, \hat{y}_i, \hat{w}_i, \hat{h}_i)$, and

$$t_{x_i} = (\hat{x}_i - x_i)/w_i \qquad t_{y_i} = (\hat{y}_i - y_i)/h_i$$
$$t_{w_i} = log(\hat{w}_i/w_i) \qquad t_{h_i} = log(\hat{h}_i/h_i)$$

the bounding box loss is defined as

$$L = \frac{1}{N}\sum_{i=1}^{N}(t_{x_i}^2 + t_{y_i}^2 + t_{w_i}^2 + t_{h_i}^2) \tag{2}$$

We add L2 regularization loss to the bounding box loss to get the total loss. We train the model to minimize the total loss.

Here, we use a "tiny" ResNet to predict the bounding box. There are 9 conv layers and 2 fully connected layers.

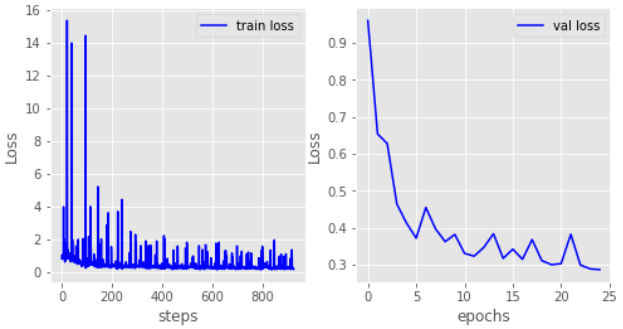


Figure 8: Training loss and validation loss of bounding box prediction

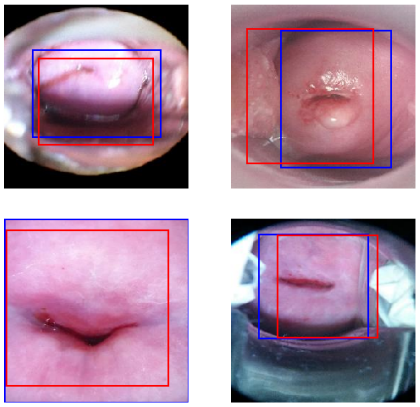


Figure 9: Bounding box prediction examples (Blue rectangle is the ground truth bounding box labeled manually, and red rectangle is the predicted bounding box from the neural network)

Compared with ResNet 18-layer in Table 1 in [8], the first conv layer is the same conv1 in Table 1 in [8], the second to fifth residual blocks are the same as conv2_x to conv5_x, except that they are multiplied by 1 instead of 2 in Table 1 in [8], so the number of conv layers decrease by half for the second to fifth residual blocks. The sizes of two fully connected layers are 512 and 4 respectively, with dropout probability 0.8 applied to the first fully connected layer. The activation function for these two fully connected layers are relu and sigmoid multiplied by 250 ((x_i, y_i, w_i, h_i) range from 0 to 255). We use learning rate to be 1e-6, l2 regularization strength to be 1e-5, batch size to be 32, and Adam optimizer to update trainable variables in the model.

Fig. 8 shows the training and validation loss histories. There are some unusual peaks in the training loss history. The reason is that some images have very small bounding box, which results in large loss even absolute difference between (x_i, y_i, w_i, h_i) and $(\hat{x}_i, \hat{y}_i, \hat{w}_i, \hat{h}_i)$ is at a similar scale as difference in images which have large bounding box. Fig. 9 shows some examples of predicted bounding box and ground truth bounding box.

6. Conclusion and Future Work

In this paper, we compare the four most popular architectures in image classification problem and we find that Inception v3 performs best compared to AlexNet, VGG16 and ResNet. The cervical images are significantly different from images on ImageNet, and therefore, we get better performance to train the model from scratch than to use pretrained model on ImageNet and fine-tune the model on the cervix dataset. Saliency map visualization prompts us to improve identification for type three cervix through extraction of Region of Interest, which we did both manually at first and later with a bounding box regressor. We also find that a "tiny" ResNet can identify the bounding box of cervix from images with a large region of irrelevant information.

Due to time constraints, not all our ideas are implemented. we think the following appraoches can potentially improve the performance of the DCNN to identify cervix types.

1. Given that we have a bounding box regressor and a CNN, We could use faster R-CNN [13] to combine them for cervix type classification. Since there are some good results from using only regions of interest to identify cervix types and from predicting bounding box, it is possible that faster RCNN could have lower classification loss and higher accuracy.

2. Spatial pyramid pooling (SPP) [7] is robust to object scale and deformation. Since the region of interest (cervix) in the cervical images varies in both size and aspect ration wildly, adding the spatial pyramid pooling layer on top of the convolutional layers could improve recognition accuracy.

3. We find the performance of DCNN is very sensitive to hyper-parameters. We could use random search for hyper-parameter optimization ([2]) to tune hyper-parameters. In addition, we could look for a better scheme to use additional training set.

4. In order to generate more features, we may use the Generative Adversarial Networks(GAN) to train the images and generate additional features and concatenate the features with the features from last fully connected layer. In doing so, we may improve on the accuracy.

References

[1] M. Arbyn, X. Castellsagu, S. de Sanjos, L. Bruni, M. Saraiya, F. Bray, and J. Ferlay. Worldwide burden of cervical cancer in 2008. *Annals of Oncology*, 22(12):2675, 2011.

[2] J. Bergstra and Y. Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305, 2012.

[3] D. C. Cireşan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber. Flexible, high performance convolutional neural networks for image classification. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Two*, IJCAI'11, pages 1237–1242. AAAI Press, 2011.

[4] D. Ciregan, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3642–3649, June 2012.

[5] D. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber. Deep neural networks segment neuronal membranes in electron microscopy images. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 2843–2851. Curran Associates, Inc., 2012.

[6] M. Havaei, A. Davy, D. Warde-Farley, A. Biard, A. Courville, Y. Bengio, C. Pal, P.-M. Jodoin, and H. Larochelle. Brain tumor segmentation with deep neural networks. *Medical Image Analysis*, 35:18 – 31, 2017.

[7] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *European Conference on Computer Vision*, pages 346–361. Springer, 2014.

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[9] K. Kamnitsas, C. Ledig, V. F. Newcombe, J. P. Simpson, A. D. Kane, D. K. Menon, D. Rueckert, and B. Glocker. Efficient multi-scale 3d {CNN} with fully connected {CRF} for accurate brain lesion segmentation. *Medical Image Analysis*, 36:61 – 78, 2017.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[12] F. Milletari, N. Navab, and S. Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. *CoRR*, abs/1606.04797, 2016.

[13] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[14] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015.

[15] M. Ross, W. Wei, and L. Ohno-Machado. big data and the electronic health record. *Yearbook of Medical Informatics*, 9, 2014.

[16] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[17] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[18] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.