Histopathologic Cancer Detection using Convolutional Neural Network

Harit Kapoor
School of Computing Science and Engineering, Vellore Institute of Technology Chennai Campus, Chennai,India
harit.kapoor3@gmail.com

Yukta Gupta
School of Computing Science and Engineering,Vellore Institute of Technology Chennai Campus Chennai,India
yuktagupta.23@gmail.com

Syed Ibrahim S P
School of Computing Science and Engineering,Vellore Institute of Technology Chennai Campus Chennai,India
syedibrahim.sp@vit.ac.in

***Abstract*—Histopathology is the microscopic study of the tissues in order to study the evidence of disease. Histopathologic cancer detection refers to the examination of the biopsy after the specimens have been place onto the glass slide. Recent advancements in deep learning-based algorithms have shown significant improvements in histological analysis of lymph node metastasis. Still the accuracy is not increased because of the difficulty in the handling of the images. In this paper, we have proposed and evaluated a Convolution Neural Network (CNN) based deep learning model to detect Cancer metastasis on histopathology images of lymph nodes to potentially improve the accuracy and efficiency of the diagnosis.**

Keywords—convolutional neural network, deep learning, histopathology

I. INTRODUCTION

Histopathologic cancer detection means to detect the presence of the metastatic cancer in the specimen of the tissue obtained from biopsy. The problem which is addressed involves classification of the slide images of the tumors. The image can be classified into two class labels either Malign or Benign. If there is a metastatic cancer cell in the image, it needs to be classified as malign else it is classified it as benign. Automated lymph node metastasis detection has a very significant potential to improve the efficiency and accuracy of pathologists for the diagnostic process in this field.

In the last few years there have been significant improvements in the field of computer vision task using Convolutional Neural Networks [1]. Following this pattern, in recent years CNN based computer aided metastasis detection have been proposed [2,3,4].

In this paper, we introduce a robust method to predict presence of lymph node metastasis from whole slide pathology images using Convolutional Neural Networks. The architecture used to train the model for the detection of the lymph node metastasis does not use any pre trained model like VGG16, NasNet, Inception or any other model. The architecture used to classify the images has been implemented from scratch using Keras.

II. RELATED WORK

The research work in the field of histopathologic cancer detection using Convolutional neural network is very vast domain. There have been a number of methods which have been proposed over the years for detection of cancer in images using CNN. The methodology proposed in paper makes use of a novel architecture rather than using pre trained models. This helps to reduce the cost of computation and results in faster convergence model. The dataset used for the purpose also packs the clinically - relevant task of metastasis detection into binary classification which can be trained very easily on a single Graphical Processing Unit (GPU) along with good accuracy scores.

III. METHODOLOGY

The methodology used for detection of the lymph node metastasis from whole slide pathology images is done using Convolutional Neural Networks. The Convolution Neural Network just like any other neural network consists of neurons with weights and biases. The inputs received by the neuron are subjected to weighted average, passed through an activation function and output is given. Unlike neural networks, which take vectors for input, the CNN takes a multi-channeled image. The CNN is able to capture both spatial and temporal features of an image using relevant filters. In other words, we can use CNN to understand the sophistication of images. CNN does the role of reducing the image into such formats which makes it easier to process, while still maintaining its features which are critical for a good accuracy. The layers involved in CNN are Convolution Layer, Pooling Layer and the Fully Connected Layer.

Convolution Layer: Involved in the operation of convolution. It uses kernel/filter to extract the features from the image. Assuming an image of dimension 32x32x3 when applied with a kernel/filter of 5x5x1.A single pixel value of the resultant image is obtained by a dot product of the filter and a small 5x5x1 chunk of the image i.e. 5*5*1=75-dimensional dot product + bias.

$$\text{Pixel Value} = w^T x + b$$

The result obtained by this dot product is a scalar value. The filter/kernel is then slided over the entire image and the resultant image is obtained.

Pooling: Pooling layer is used to reduce the spatial size. This helps to reduce the number of parameters and the computation in the neural network. The pooling layer works independently on each feature map. Max pool takes the filter and stride size and returns the maximum pixel value in the maximum in that particular stride.d
Fully Connected (FC) Layer: FC Layers are used to detect specific global configurations of features detected by the lower layers in the neural net. Each neuron of the FC layers has its own weights. It connects every node from one layer to every node of another layer.

Activation Layer: It is generally put at the end of a layer or in between the neural network. The main task of the activation layer is to decide whether the neuron gives output or not. It is a nonlinear transformation that is done on the input signal to send as an output to the next layer of neurons as input. RELU is the most used activation function which converts all the negative inputs to zero and the neuron does not get activated. This makes the computation more efficient, which results in fast convergence of the neural network.

IV. Dataset Description

The Dataset used for training the model is the PatchCamelyon (PCam) dataset [5,6] which is taken from Kaggle. The PatchCamelyon is a new and challenging image classification dataset. The dataset available on Kaggle does not contain duplicate images which are present in the original data due to its probabilistic sampling. The dataset used for training the neural network consisted of 220,025 colour images. Each image is of size 96x96x3px. The images of the dataset are annoted with a binary class label which indicates the presence of metastatic tissue in the image. Out of the total training data, the number of positive samples is 89,117 and the number of negative samples is 130,908

The dataset is divided into a training set of 176,020 examples and a validation set of 44,005 examples. We have used another 57,458 images as test sample A positive label for the image indicates that the center 32x32px region contains at least one pixel of tumour tissue. The outer region is provided to enable the design of the models that do not use any zero padding.

V. Experiments and Results

In this section, we evaluate the propose architecture for the histopathologic cancer detection. The architecture of the model is shown in the table below (CN - Conv2D layer, BN - Batch normalization layer, ACT - Activation layer, MP - Max Pooling layer, DP - Dropout layer, F - Flatten layer, D - Dense layer)

Layer	Output	Params
CN	(None,94,94,64)	1792
BN	(None,94,94,64)	256
ACT	(None,94,94,64)	0
CN	(None,94,94,64)	36864
BN	(None,94,94,64)	256
ACT	(None,94,94,64)	0
CN	(None,94,94,64)	36864
BN	(None,94,94,64)	256
ACT	(None,94,94,64)	0
MP	(None,45,45,64)	0
DP	(None,45,45,64)	0
CN	(None,43,43,128)	73728
BN	(None,43,43,128)	512
ACT	(None,43,43,128)	0
CN	(None,41,41,128)	147456
BN	(None,41,41,128)	512
ACT	(None,41,41,128)	0
CN	(None,39,39,128)	147456
BN	(None,39,39,128)	512
ACT	(None,39,39,128)	0
MP	(None,19,19,128)	0
DP	(None,19,19,128)	0
CN	(None,17,17,128)	294912
BN	(None,17,17,128)	1024
ACT	(None,17,17,128)	0
CN	(None,15,15,128)	589824
BN	(None,15,15,128)	1024
ACT	(None,15,15,128)	0
CN	(None,13,13,128)	589824
BN	(None,13,13,128)	1024
ACT	(None,13,13,128)	0
MP	(None,6,6,128)	0
DP	(None,6,6,128)	0
F	(None,9216)	0
D	(None,256)	2359296
BN	(None,256)	1024
ACT	(None,256)	0
DP	(None,256)	0
D	(None,1)	257

Total params: 4,284,673

Trainable params: 4,281,473

Non-Trainable params: 3,200

We have used the above-mentioned architecture to train the model. We have introduced Batch Normalization before activation layer to address the problem of internal covariance shift. The Internal covariance problem occurs when the input distribution of the model keeps fluctuating. Batch Normalization layer helps to control the mean and variance of the output and does not let the model to over-specialize in one region of the input distribution. Using Batch Normalization, we reduce the chances of overfitting of the training example. The dropout layer is added to further reduce the overfitting as it is also a regularization technique [7]. It is a technique in which randomly selected neurons are ignored/made dead during the training. By using the dropout layer in the neural network, the network becomes less sensitive to specific weights of neurons. This results in better generalization and reduces the chances of overfitting of training data. The rate for dropout is set at 0.2 i.e. dropping 20% of the nodes randomly. The dense layer of 256 neurons is added as a FC layer. The dense layer is a FC layer which each neuron is connected to every neuron of the next layer. The final dense layer has only one neuron. The activation function used for the training of the model are RELU and sigmoid. RELU is used in the hidden layers, while sigmoid activation function is used as an activation function of the output layer.

The model was trained for 10 epochs on GPU enabled Kaggle kernel with 13 Gigabytes RAM and Nvidia K80 Graphical Processing Unit (GPU). The training and validation metrices obtained for each epoch are shown in the table below.

	Training		Validation	
Epoch	**Accuracy**	**Loss**	**Accuracy**	**Loss**
1	0.8496	0.3557	0.7992	0.5240
2	0.8982	0.2560	0.8809	0.2863
3	0.9140	0.2218	0.9263	0.1994
4	0.9238	0.2008	0.8567	0.3280
5	0.9304	0.1835	0.8774	0.2966
6	0.9357	0.1712	0.9321	0.1792
7	0.9397	0.1610	0.9224	0.2010
8	0.9435	0.1527	0.8327	0.6056
9	0.9466	0.1445	0.8904	0.3210
10	0.9487	0.1383	0.9105	0.2165

The loss metrics used is Binary cross entropy. The optimizer used is Adam optimizer with learning rate of 0.001. The test data was used to validate the accuracy of the model. The accuracy achieved by the neural network model on the testing data is 95.34%.

VI. Conclusion

In this paper, we propose a novel convolution neural network architecture for detection of metastatic cancer tissues on histopathologic images of lymph nodes. We exploited various machine learning techniques in order to increase the accuracy as well as the computation efficiency of the neural network. The experimental results gave a very good accuracy for the testing example. One direction of future work could be implementation of Rotation invariant CNN for detection of the presence of lymph node metastasis.

References

[1] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 770–778

[2] Paeng, K., Hwang, S., Park, S., Kim, M.: A unified framework for tumor proliferation score prediction in breast histopathology. In: Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support. Springer (2017) 231–239

[3] Liu, Y., et al.: Detecting cancer metastases on gigapixel pathology images. arXiv preprint arXiv:1703.02442 (2017)

[4] Bejnordi, B.E., et al.: Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer. Jama 318(22) (2017) 2199–2210.

[5] B. S. Veeling, J. Linmans, J. Winkens, T. Cohen, M. Welling. "Rotation Equivariant CNNs for Digital Pathology". arXiv:1806.03962.

[6] Ehteshami Bejnordi et al. Diagnostic Assessment of Deep Learning Algorithms for Detection of Lymph Node Metastases in Women With Breast Cancer. JAMA: The Journal of the American Medical Association, 318(22), 2199–2210. doi:jama.2017.14585

[7] Srivastava, et al. Dropout: A Simple Way to Prevent Neural Networks from Overfitting