Detecting Pneumonia from X-Ray images: Machine Learning Approaches

Sanchayan Sarkar, Zhang Lu

Abstract: Detecting pneumonia from X-Ray images is a challenging task because of high variability amongst experts, time and cost. Automatic detection of pneumonia from Chest X-ray images is required to ensure scalability of such a process. In this project, we explore various methods to detect pneumonia presence from images. We start from classical computer vision features like LBP and SURF features to state of the art features derived from Deep Convolutional Neural Networks and Inception Networks. We also use linear decision boundary classifiers like Logistic Regression and Support Vector Machines to deep nets like Inception Net and Residual Neural Network. In this report, we will show the high benefits of using deep features with near perfect precision and recall scores for our deep learning methods.

Keywords: Deep Learning, Pneumonia, CNN.

INTRODUCTION

Millions of people suffer from Pneumonia in the world. In the US alone, around 50,000 people dies from pneumonia every year [1]. Chest X-ray images are a very good indicator in detecting pneumonia and epidemiological studies. Also, Chest X-ray images are widely available. However, detecting pneumonia is challenging as it requires highly skilled radiologists, thereby involving high cost and more time. Also, availability of such radiologists might not be that frequent enough. Automatic detection of pneumonia from such X-ray images therefore becomes a natural solution. However, such an automatic detection is not easy as their appearance in X-ray images is often ambiguous and can overlap with other diagnoses and abnormalities. Figure 1 shows a example of a Chest-X ray image and it's pneumonia counterparts.

Recently, deep learning and artificial intelligence approaches have made serious progress in medical computer vision [2]. In this project, we are going to state the various methods we have built, modified and used in solving pneumonia detection binary classification problem. The rest of the paper is organized as following: *Related Work, Methodologies, Results, Discussion* and conclusion.

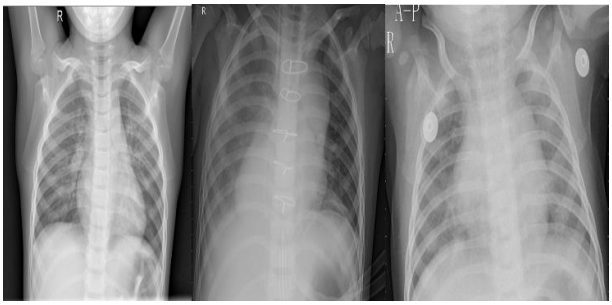

Figure 1 Left : Normal X-ray image Middle: Viral Pneumonia Right: Bacterial Pneumonia

RELATED WORK

In the early days of computer vision, hand crafted feature extraction methods were the most common approaches that was sought after like image segmented objects and identifying these segments using statistical classifies [3]. Approaches like LBP [11], HOG [12] and SURF have been proved to be very powerful feature extraction methods since they have been proposed. Inspired from them, we have used two approaches in our method as baselines.

Of late, advances in deep learning has tackled some of binary classification task of detecting pneumonia. Deep convolution neural networks like the Alexnet [5] VGGNET [4] have been very successful methods in large scale image classification challenge like ImageNet. They have also been finetuned and used in large scale medical datasets like Chest X-ray 14 which has around 100000 images. Within a few years of these convolution neural networks, different variants like the InceptionNet or GoogleNet [6] were published uses a inception module to stack up multiple convolution units without affecting the accuracy. It also beat the ImageNet challenge. Currently, even deeper layers are being conceived, like the Residual Neural Network (ResNet) [7] and its multiple variants. It uses a residual block that allow higher non linearities in the model without affecting the training accuracy. It also enables to train deeper layers. State of the art architectures are as deep as 121-layered Convolutional Neural networks like the ChexNet [8] which have been used for the problem of pneumonia detection. Inspired from this we have selected InceptionNet and ResNet-50 model for modifying and re-training on our dataset. We also developed our own vanilla Convolution Neural Network that is trained from scratch with our dataset. The following section walks through the methods in detail.

METHODOLOGY

Handcrafted features and Shallow Methods

Classical Features:

Local Binary Patterns (LBP)

We applied LBP to describe the texture of our input X-ray image. In order to construct the LBP texture descriptor, we need to convert each pixel in the image in to a set of binary digits according to the neighbors of each pixel. We applied the basic LBP (using 3x3 neighborhood). First, we use the top, bottom, left, right and four corners' pixels to compare with the center pixel one by one. If the center is larger, we assign a '1' to its position, otherwise, assign an '0'. After preform LBP to image, all pixels will become an integer number between 0 and 256, because it uses 8 neighbors to calculate, so the length of the binary representation would also be 8. Then we flatten the full LBP 2D array representation of the input image and get the feature we want.

Histogram of oriented gradients (HOG)

The HOG descriptor essentially wants to represent the local object appearance and shape with the distribution of intensity gradients or edge directions. First, we focus on small regions and only look at the pixels inside individual region, calculate the histogram of gradient directions. Then, concatenate these histograms of different regions.

HOG have some advantages over other feature descriptors. Basically, it's a local descriptor because it focuses on small region of image rather than the whole image, which gives the feature more locality and makes it more robust to clutter and occlusion. Also, it has good repeatability and flexibility because it is invariant to geometric transformations and intensity variation. Overall, the characteristic of HOG makes it a good descriptor to our task of processing X-ray image.

Classifiers

Logistic Regression [14]

Logistic Regression is a widely used model for classification problem. The features of

our data are discrete and continuous, if we want to analyze whether a data should be positive sample or negative sample Binary logistic regression is the model we want. Logistic regression model relates the label (positive or negative) to the features we extract in previous step. It is a linear method, but the predictions are transformed using the logistic function. The impact of this is that we can no longer understand the predictions as a linear combination of the inputs as we can with linear regression. Also, Logistic regression model is naturally capable to handle non-linear relationships between the dependent variables and independent variables, since it applies a non-linear log transformation based on the linear regression. After training, the model will learn the logit transformation. The model predicts which class a data belongs based on a threshold and it's also named decision boundary.

Support Vector Machines [15]

SVM is a linear model which have remarkably robust performance in classification problems. The main idea of SVM is that the model wants to find the hyper-plane which can separate the data into classes by maximizing the margin with support vectors, which means it tries to make a decision boundary that separate the classes as wide as possible. It can find hyper-plane in feature space which is equivalent to the decision boundary in input data space. In non-linear separable tasks, SVM can work in combination with 'kernels' which introduce a non-linear mapping to feature space.

Deep Learning Approaches

Convolution Neural Network (CNN)

We have developed a convolutional network architecture having three convolution layers and two dense layers. Figure 2 shows our network architecture. The convolution layers learn different filters over the input image finding out different layers of spatial features over the input images.

Layers	Neurons	Activation
Conv1	3x3x32	RELU
Conv2	3x3x32	RELU
Conv3	3x3x64	RELU
Dense	1x64	RELU
Dropout	-	-
Dense	1x1	Sigmoid

Figure 2 Our CNN architecture

The dense layers are fully connected hidden layers (all weights are connected to the previous and next layer). The output layer consists of only 1 neuron as it will hold the probability of the outcome between 0 and 1.

We have used RELU activation for all our layers because it captures high non linearities. The final layer has a sigmoid activation to give the class probability. The dropout ratio is 0.5 which means 50% of the dense neurons will be dropped. This is to reduce the chance of overfitting.

We used the RMS Prop optimization algorithm for backpropagation which reduces gradient variability across batches by regularizing it with the root of the squared gradients. The following equations shows the basic structure of this optimization process.

$$E[g^2]_t = \beta E[g^2]_{t-1} + (1-\beta)\left(\frac{\delta C}{\delta w}\right)^2$$

$$w_t = w_{t-1} - \frac{\eta}{\sqrt{E[g^2]_t}}\frac{\delta C}{\delta w}$$

η is the learning rate and β is the hyperparameter. C is the cost function which in our case is the binary cross entropy loss mentioned below.

$$C(w) = -t_1 \log(s_1) - (1 - t_1)\log(1 - s_1)$$

The RMS prop optimization ensures faster convergence which is key because we have a smaller dataset and less resources to work with. We trained this model from scratch on our Chest-X Ray dataset which is mentioned in the following section.

Convolution Neural Network as a Feature Extractor + SVM

For this, we used the VGG Net [4] network as a feature extractor. This network is already trained on ImageNet dataset. So, we took the "FC2" layer of the network as a feature embedding for our data. We then trained a Linear SVM on top of it. We did not finetune this network on our data but this was a much bigger architecture and could have easily overfit on our data. We obtain some good results which are given in the following section.

Transfer Learning from Inception-Net Version3

Deep convolution networks have a problem when it comes to stacking deeper layers. The number of parameters become too large. In figure 3, the convolution layer has to be (5x5x192) and the number of parameters will be (5x5X192) x (28x28x32) =120 million which is obviously huge. To avert this problem, the Inception introduces something called bottleneck which is shown in Figure 4.

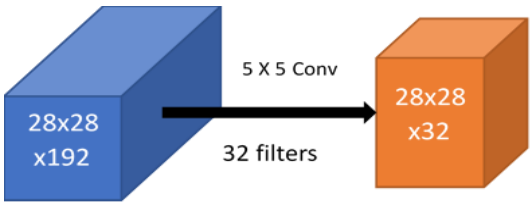

Figure 3 A CNN block

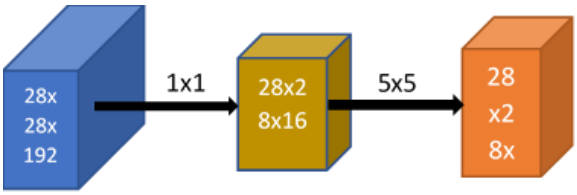

Figure 4 Inception Module

Because of this bottleneck through a 1x1 convolution filter the number of parameters to be learnt is (2.4+10) =12.4 million. This is 10x performance improvement.

Layer	Structure/stride
Conv	3x3x32/2
Conv	3x3x64/1
Padding	3x3/1
Pool	3x3/2
Conv	3x3x80/1
Conv	3x3x192/2
Conv	3x3x288/1
3xInception [6]	Figure 5 of [6]
5x Inception [6]	Figure 6 of [6]
2x Inception [6]	Figure 7 of [6]
Pool	8x8
Dense	2048
Dropout	-
Avg. Pool	-
Dense	128
Dense	2

Figure 5 Modified Inception Net Version 3

This allows for deeper networks without learning too many parameters. In our problem, we have used the Inception Net (v3) [6] which has over 5 inception modules stacked. We modified it and our version is shown in Figure 5. We used RELU activation for all layers except the last where we used Sigmoid activation. For finetuning, we copied the weights of all the layers except the Dense layers. This will make sure we don't

update the weights of earlier layers as they are already trained on ImageNet dataset. Categorical Cross Entropy Loss was used as the objective function with ADAM optimizer.

Residual Neural Network (ResNet50)

Deep stacking of convolution layers has a degradation problem where the optimizer after a long number of iterations begins to diverge resulting in a higher training loss [7]. To overcome this, a residual block were introduced (Figure 6), where the network learns the residual mapping rather than the actual function.

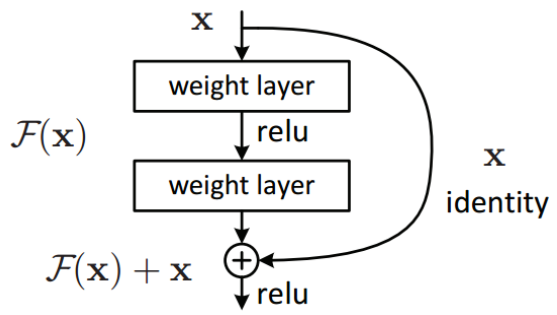

Figure 6 Residual Block

The identity is skipped for over two layers and added to the previous activation, i.e. F(x)+x. Because of this, x is easily identifiable and this ensures one can stack as many layers as possible without degrading the training loss or overfitting. It also introduces high number of non-linearities to the model. In our problem, we used the ResNet50 architecture and finetuned it to our dataset. The architecture is given below.

Layer	Structure/Size
Conv1	7x7x64/2
Max-pool	3x3/2
Conv2_x	$\begin{pmatrix} 1x1x64 \\ 3x3x64 \\ 1x1x256 \end{pmatrix} x3$
Conv3_x	$\begin{pmatrix} 1x1x128 \\ 3x3x128 \\ 1x1x512 \end{pmatrix} x4$
Conv4_x	$\begin{pmatrix} 1x1x256 \\ 3x3x256 \\ 1x1x1024 \end{pmatrix} x6$
Conv5_x	$\begin{pmatrix} 1x1x512 \\ 3x3x512 \\ 1x1x2048 \end{pmatrix} x3$
Avg. Pool	-
Dense	1000
Dense	1

Figure 7 ResNet50 Configuration (Modified)

We just added an extra dense layer with sigmoid activation function. The second last dense layer was also trained. All the convolution layer weights were copied as the network was pre-trained on ImageNet. We used Cross Entropy loss function with ADAM optimizer.

EXPERIMENT & RESULTS

Dataset

We used the dataset Chest X ray from Kaggle [9]. It has 5217 Training images and 624 test images. It has 3875 images of people infected with Pneumonia and 1875 normal images. It has two types of pneumonia categories but in this problem, we don't consider that because it is a binary classification task.

Data preprocessing

For all of the methods, we did image mean normalization and for the deep learning methods we also did batch normalization. Images were resized to the appropriate input sizes for the respective deep learning networks. Data augmentation was done for

the deeper approaches by scaling + rotating for the Resnet and flipping it horizontally.

Evaluation Metrics

For evaluation we used Precision, Recall and the Accuracy on the training set. Since, the data split was already done in Kaggle, all we did was random shuffling.

Results

The results for all of the approaches are given in the following Figure.

Method	Precision	Recall	Accuracy
LBP+LR	0.61	0.68	0.65
LBP+SVM	0.62	0.66	0.68
HOG+LR	0.68	0.72	0.75
HOG+SVM	0.71	0.79	0.78
CNN (scratch)	0.79	0.88	0.84
VGG +SVM	0.77	0.87	0.83
Inception	0.86	0.95	0.94
ResNet50	0.81	0.98	0.94

DISCUSSION

As we can see from the results, all of the deep learning methods does considerably better than any of the classical methods with linear classifiers.

One of the interesting findings from this project is that our architecture does fairly well but it tends to overfit a bit since the training loss obtained was around 0.974 after 20 iterations. This can be easily explained because our dataset is considerably small compared to the number of parameters that are learnt when it is trained from scratch.

Other than that, we see that transferring learning really helps in increasing the accuracy. Both the InceptionNet and the Resnet50 has staggeringly high recall rates. […….]. What makes is more interesting is that all of these finetuned networks were trained on natural images from ImageNet and they all work with medical images. An explanation can be that the low-level features are quite common between medical images and natural images. It would be interesting to see a heatmap as to what features are getting learned.

This also proves that creating deeper nets using Residual learning and stacking layers actuallymakes the model more robust because of its' high non linearities. This opens up new possibilities in the field of medical vision.

CONCLUSION

In this project, we have tried multiple approaches in detecting pneumonia from Chest X-Ray image. We have classical computer vision features like LBP and HOG with shallow trainers like Logistic Regression and SVM. We also tried state of the art deep learning approaches like Convolution Neural Networks, VGG net, Inception Net and ResNet and found considerably better results. From this we conclude that as the size of the data set increases, deep learning approaches will fare much better. However, the issue of interpretability still remains an issue when it comes to medical science. In future, we would like to train networks like Siamese Networks by learning similarities and dissimilarities between multiple categories of pneumonia. We would also like to train our existing methods on a larger dataset and see what features are getting learnt.

ACKNOWLEDGEMENT

We would like to thank Dr. Milos Hauskrecht for giving us the opportunity to do this term project. Also, we would like to thank [10] for our implementations with the deep learning.

References

[1] URL https://www.cdc.gov/features/pneum56onia/index.html.

[2] Klang E. Deep learning and medical imaging. *J Thorac Dis*. 2018;10(3):1325–1328. doi:10.21037/jtd.2018.02.76.

[3] Kermany, Daniel S., et al. "Identifying medical diagnoses and treatable diseases by image-based deep learning." *Cell* 172.5 (2018): 1122-1131.

[4] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556* (2014).

[5] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems*. 2012.

[6] Szegedy, Christian, et al. "Rethinking the inception architecture for computer vision." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

[7] He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

[8] Rajpurkar, Pranav, et al. "Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning." *arXiv preprint arXiv:1711.05225* (2017).

[9]https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia

[10] https://github.com/deadskull7/Pneumonia-Diagnosis-using-XRays-96-percent-Recall

[11]https://www.kaggle.com/curiousprogrammer/chest-x-ray-image-classification-tf-hub-resnet50

[12] Liao S, Law M W K, Chung A C S. Dominant local binary patterns for texture classification[J]. IEEE transactions on image processing, 2009, 18(5): 1107-1118.

[13] [10] Dalal N, Triggs B. Histograms of oriented gradients for human detection[C]//international Conference on computer vision & Pattern Recognition (CVPR'05). IEEE Computer Society, 2005, 1: 886--893.

[14] Bishop C M. Pattern recognition and machine learning[M]. springer, 2006.

[15] Furey T S, Cristianini N, Duffy N, et al. Support vector machine classification and validation of cancer tissue samples using microarray expression data[J]. Bioinformatics, 2000, 16(10): 906-914.