**A Novel Architecture for Feature Extraction and Pathology Detection from Chest X-Ray Images**

*N.Sarada*1\**, Dr. K. Thirupathi Rao*2

*1 Research Scholar, Department of Computer Science & Engineering, K L E F, Guntur, India*

*2 Professor & BOS Chairman, Department of Computer Science & Engineering, Associate Dean-Academics, K L E F, India*

**Abstract:** Neural networks are widely used for the automation of analysis and classification tasks in the fields of medical image processing. They have successfully achieved state of the art performance in medical image segmentation and feature extraction techniques. This automatic classification in the medical field is very helpful in developing tools for early detection of dreadful pathologies like tuberculosis and pneumonia in areas where access to doctors or radiologists in scarce. In this work, we propose a novel approach for the classification of lung pathologies like tuberculosis and pneumonia by masking them in boundary boxes using convolutional neural networks. Our solution provides a flexible way by using saved trained models that could be directly employed by the Radiologists. In this paper, we describe the architecture required to achieve such a scalable model which could be used by doctors and radiologists without too much training in technology. The proposed convolutional architecture consists of connected components which are parallel residual blocks and sampling layers. The images do not lose their original quality, giving the best error free predictions. We envision this model to be deployed in labs, providing access to medical imaging expertise in some of the most remote places in the world.

# Introduction and Related Work

Identifying the pathologies is one of the challenging tasks in medical fields, it’s still an epidemiological study. More than one million people get hospitalized and more than 50,000 people die with tuberculosis. Diseases like these are reported to increase at a very unusual rate due to human immunodeficiency virus, bacterial immunity to medication, a rising number of the homeless/drug users, and so on [1, 2]. Tuberculosis is an infectious disease that affects the lungs caused by Mycobacterium, doctors term these infected, affected regions to be bacillus. If a patient is diagnosed with bacillus in lung x-ray, it will be termed as tuberculosis positive/positive spectrum smear. According to the World Health Organization, if one bacillus is present in 100 images, a minimum of 100 fields are examined to declare the positive sputum smear [3]. To identify this bacillus or the affected areas in the lungs, we need to be precise in color, shape, and edges. This can be extracted using image processing or pattern recognition techniques. Previously, research scientists used traditional machine learning and computer vision techniques like support vector machines, multi-layer neural networks that dealt with many expenses to detect the spectrum spear. The CAD (Computer- aided diagnosis) is used before machine learning algorithms [4] which follows four steps in finding the anomaly of the given CXR images. These include pre-processing, segmentation, feature extraction, and classification. However, CAD systems only use computer vision techniques in finding the affected regions, they couldn’t beat the human level accuracies. Later in the year 2012, when deep learning has been popular due to its end to end architectures, it could extract features of how pathologies could be identified in the given CXR images.

Manual screening for identifying the presence of the bacillus or any pathologies in lungs consumes much human power which isn’t accurate and reliable. Hence we need an automated process to analyze the abnormalities present in the lungs from a given x-ray image of the lungs with less or no human interference. Ginneken et al. [3] is the one who proposed the automatic detection of tuberculosis using

K-nearest neighbors. In his method using KNN[5], the primary metric for the classification is the AUC value, which is the area under the curve, giving a value of 0.82 on 388 images of TB data set and 0.98 on 200 images of lung pathologies. This score is derived from the combination of the intersection of different detected regions of the overall system. Later in the year 2010, Hogeweg et al. [6] pro- posed a method which is an advancement to the KNN algorithm, in his research a group of images is divided into circular regions, and by using a classifier called Linear Discriminant Analysis, the texture score of the given CXR images are detected pixel-wise.

In the year 2012, Jaeger et al. [7] used manual masks to identify the affected regions of lungs by segmentation. Once the images have segmented the features like shape, the texture is used to find the pathological pattern in the given input images. To represent these descriptors and their distributions visually, histograms and boxplots are used, and when the patterns are extracted from the CXR images, they have used a support vector machine algorithm to classify them into positive and negative detection of abnormality. Later in the year 2014, Jaeger et al.[8] proposed a similar method using support vector machines, wherein they have used two feature sets, one object detection based and the other content-based image retrieval systems. These gave a top-notch performance when considering the AUC curve metric which is found to be 0.87 and 0.90 respectively. Later in the year 2016, Karargyris et al. [8] proposed a method to identify pulmonary abnormalities using a support vector machine classifier. This improved the accuracy of the AUC curve by 12% which is to 0.934. This method is faster and cheaper when compared to previous classification algorithms used. Maduskar et al. [9] also introduced an algorithm for finding pleural effusion, this could be able to extract features such as angle, intensity, and other biological information from the given CXR image. The algorithm used is a random forest classifier which is trained over a thousand images, the metric considered is the AUC curve which returned an accuracy of 87%.

In this work, we will be using Convolutional Neural Networks to classify the given lung pathologies based on their classes whether they fall into normal or abnormal. The architecture can be generic to Tuberculosis and Pneumonia pathologies. It mainly relies on the hospital dataset which we will be training using ConvNets to extract the features and identify the masks (the affected regions of the lungs either bacillus or the positive pneumonia region) via segmentation. The neural network is given masks of pneumonia affected areas and the CXR images. The model is trained over a set of images; once the training process is complete, the weights will be able to predict if the pathology is positive or negative in a given model by masking the affected region.

# Convolutions for Image Segmentation

* 1. *Basic Architecture and Layers*

Medical image segmentation plays a significant role in many image applications to derive interesting results. We have various architectures to segment the images. Convolutional Neural Net- work (CNN) architecture is the most popular with higher accuracy. These are based on an idea of local spatial connectivity and sharing of weights among the layers. Here, a brief background knowledge would be provided for CNNs.

Convolution is a mathematical operation on two functions, which gives the integral to the pointwise multiplication and one of the original functions will be translated.

(1)

CNNs operate over volumes of data which assume that input is a multi-channeled image. It arranges its neurons in three dimensions, width, height and depth, hence every layer transforms 3D input volume to 3D output volume.

A CNN has three layers, a Convolutional layer, Pooling layer and a Fully-connected layer.[10]

**Convolutional layer**: It consists of filters, which extracts the information and features present in the image. It calculates the dot product of the input image with filter, later an activation function, say ReLU is applied to it. And this is passed through the pooling layers. Later, the same set of operations will be repeated again.

**Padding**: It helps in taking the whole image instead of confining to a specific part of an image. When padding is applied, the feature map (filter convolved with input tensor) would be a matrix or a tensor of size *n* + 2*p f* + 1, where n is the size of an input image, fix the filter size and p is the padding size. If there is no padding, the matrix or a tensor will be a size of *n f* +1.

−

−

**Stride**: It defines the number of pixels that could be skipped while convolving the filter with the input image tensor. The size of the matrix or tensor can be defined by the formula,

*n*+2*p*−*f*+1*.* (2)

*s*

**Pooling layer**: It helps in optimizing the output and downsampling of the features, typically applied after convolutional layer. Max and Average pooling are the two pooling operations where max pooling is a sample based discretization process and average pooling involves in calculating the average for each patch of the feature map which is down Sampled to the average value.

**Fully-connected layer**: It operates on flattened input where every input is connected to every other neuron in the following layer. Later, a softmax activation function is applied to get the class scores.

Let the input signals be x, then the following layers *xj* would be,

*Xj* = *ρ*(*Wj*)(*xj−*1) (3)

where *Wj* is a linear operator and *ρ* is to depict the non-linearity, that is the activation function[11]. Each layer can be written as the sum of convolutions of the previous layers,

*xj*(*u,kj*)=*ρ*(∑(*xj−*1(*.,k*)∗*Wj,kj*(*.,k*))(*u*)) (4)

where \* is the convolution operator.

**ReLU activation function** is defined as, R(z) = max(0, z), when z>0,elseR(z)=0,when z=0.It is the most used in CNNs.

**Softmax activation function** can be mathematically defined as,

It is a logistic activation function used for multi-class classification.

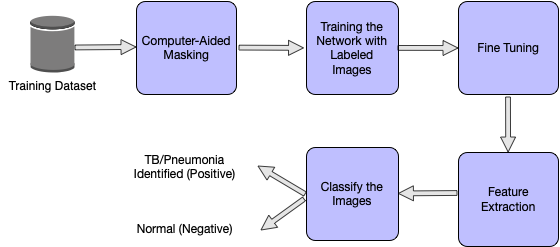
**Dropouts** could also be used, after the pooling layers. It avoids the overfitting problem, where the model learns too much out of the input data fed into it. It randomly drops neurons along with their connections from the neural network during training. This prevents units from co-adapting too much. [12]

# Proposed Approach

* 1. *Segmenting the masked regions*

Segmentation is the partitioning of an image into various segments which are a set of pixels. It is based on the similarity of pixel components, the pixels which are similar are grouped into one cluster. It can be implemented using clustering, thresholding, edge-detection, etc. These conventional techniques give out lesser accuracy in comparison to Artificial Neural Networks which has given scope for many novel approaches such as using Convolutional Neural Networks to perform image segmentation.[13] CNN produces heat maps or masks as outputs when an input image is given to it. Regions of high similarity among the adjacent pixels are formed. The network should be initially trained with a set of input images and their masks, and later heat maps or masks would be generated for the test images.

In our proposed approach, we are using a Convolutional neural network to segment the given chest radiographs (CXR) to detect pneumonia.



**Fig. 1**: Phases of the proposed approach

* 1. *Dataset Overview*

We use RSNA pneumonia dataset, which is divided into both, train and test image samples. The training labels are also provided which tells the coordinates of the mask in the images. All the train images are in DICOM format. The train labels are provided with a set of patients and bounding boxes. The bounding boxes are defined as, x-min y-min width height. There is a binary target column Target, indicating Pneumonia or Non-Pneumonia and there may be multiple rows per Patient ID. There are 26,684 input CXRs and 3000 test CXRs. The train labels file comprises the filename, and the pneumonia locations which are transformed into a dictionary wherein the filename acts as the key and the pneumonia locations are the values. This dictionary only has file names where pneumonia is detected, and if there is no pneumonia in that file, it is not included in the dictionary. Out of 26,684 images, 6012 files have shown a positive sign regarding the presence of pneumonia.

* + 1. *Dataset generator:* The dataset is broken down into random chunks to feed these set of images and masks into the model. Along with this, a few parameters, say, the batch size, image size, if the images are to be shuffled, etc are also given to the model. The images are resized to 256 x 256 from the original 1024 x 1024 and the default batch size used is 32. These images or CXRs are then sent to the Convolutional Neural Network to train the model.
  1. *Proposed architecture for segmentation*

The Convolutional Neural Network is trained such that different parameter values are optimized to extract useful features from the given input CXRs. The bounding boxes are drawn around the predicted areas of pneumonia. The network is initially given the training images and labels. CNNs then extract low-level features followed by high-level features.

The proposed CNN has residual blocks with down sampling and convolutions. An up sampling layer is used at last to resize the image to the original shape

*3.3.1 Residual block:* Residual block is just a stack of layers,

Wherein the output of the previous layer is connected to the output of new layers. This increases the accuracy of the model, by repeating a set of Convolutional-Batch-Normalization [14]-Activation layers. In this residual setting, it is not just passing the output of layer 1 to layer 2. Instead, it adds up the outputs of layer 1 to the outputs of layer 2. [15] In a conventional set of layers, the output y will be f(x). *y* =*f* (*x*) But in a residual block, the output would be, *y* = *f* (*x*) + *x*

* + 1. *Downsampling:* Downsampling reduces the size of the input image similar to the pooling operation. In our paper, the inputs are passed to Batch Normalization-Activation-Convolutional- Max Pool layers to perform the down sampling operation. It is a crucial operation in extracting the features from the images. Batch normalization is a technique for training very deep neural networks that standardizes the inputs to a layer for each mini batch. This has a effect of stabilizing the learning process in drastically reducing a number of training epochs require to train network.
    2. *Upsampling:* Upsampling layer brings back the resolution of the down sampled information. It restores the image to its original resolution using interpolation. After the features are extracted, we need to find the original image which is segmented to detect and depict if there is any abnormality in the chest X-rays. To accomplish this, we use an Up sampling layer.

The proposed CNN architecture goes like this, the input image of size 256 x 256 is passed to the conv2d layer having 32 random filters with a size of 3 x 3, this gives 32 feature maps of size 256 x 256 x 1. Later, this image is down sampled using Batch-Normalization- Activation (Leaky ReLU)[16]-Convolutional-Max Pool layers by a factor of 2 giving an image of size 128 x 128 x 64. To incorporate non-linearity, we are using Leaky ReLU. It solves the problem of vanishing gradients. The number of filters taken for every down sampling and the residual block is doubled, therefore, in the next pass, the number of filters would be 128. This helps in performing an in- depth analysis of the input image. Next, this output is passed onto the residual blocks wherein Batch-Normalization, Leaky ReLU and convolutional operations are performed twice, adding the output to the input at last, which is the actual advantage of the residual block. We have taken 4 blocks performing down sampling operation, and for every down sampling operation, 2 residual blocks are created. After, a set of the above operations are performed and it down samples until the image diminishes to 16 x 16. To get the image back to the original size, an up sampling operation is done which gives the output of image size 256 x 256. Layer-wise architecture is presented in the in Fig2 and a detailed explanation of each layers given in Table1.

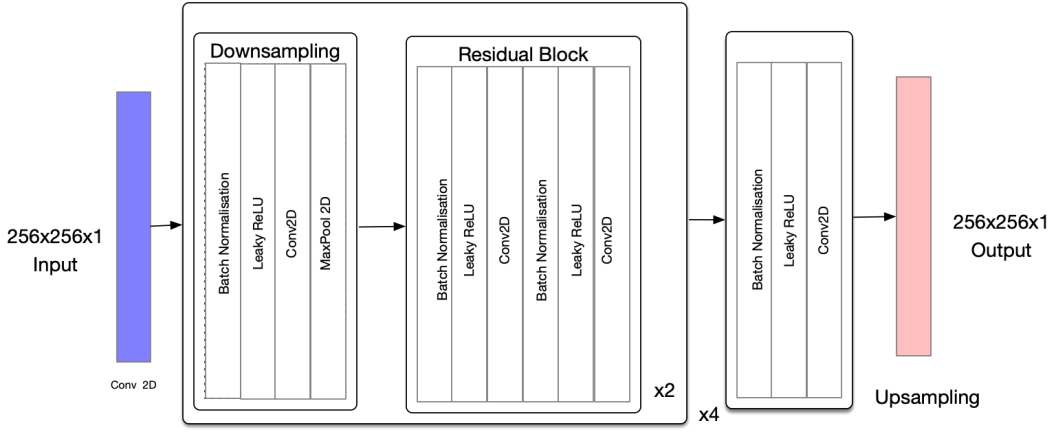
* + 1. *IoU or Jaccard loss function:* Intersection over Union metric, also called as the Jaccard Index is a method to quantify the overlap between the target image and the predicted output. It is equal to the number of pixels common between these two images divided by the total number of pixels present across both of them.

*target* ∩ *prediction*

*target* ∪ *prediction*

The IoU is calculated over each class separately and then averaged over all classes to provide a global, mean IoU score.

* + 1. *BCE Loss:* BCE is the binary cross entropy which we typically use in a binary classification problem. In our paper, we are



**Fig. 2**: Architecture of the Proposed CNN

detecting if the person has pneumonia or not. Therefore, there are two classes to be dealt with. BCE can be defined as,

Where *y* is the given label and *y*ˆ(*i*) is the observed label.

These two losses are combined to account for the total loss (IOU BCE loss) incurred in the network and mean IoU would be used as a metric to compile the model.

* + 1. *Cosine Annealing:* Cosine annealing can be alternatively called as Stochastic Gradient Descent with Restarts. [17] It decreases the learning rate in the form of half a cosine curve. This is useful because at first, we can start out with higher learning rates and later decrease as we get closer to the local minimum. This helps in exploring the deeper, but narrower parts of the loss function and has shown good results in comparison to the case where the learning rate is constant. We have used this formula for updating the learning rate,

*Lr* = *lr* ∗(*cos*(*π*∗*x/epochs*)+1*.*)*/*2 (8)

*π x/epochs* restricts the cosine range from -1 to 1. 1 is added to it which gives the range varying from 0 to 2. Now, this is divided by 2 which halves it to revert it back to 0 to 1. This way, the learning rate decreases from the max value and then again restarts with the maximum at the beginning of the next cycle.

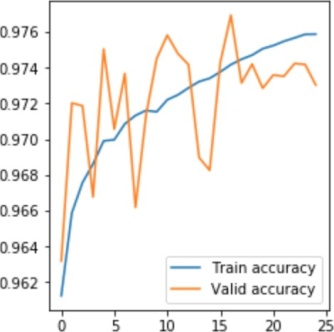
∗

# Network Training

A neural network model entirely relies on the metrics that we con- sider for the training. In the literature, we have seen many training models which are used for the segmentation process. Few architectures like LeNet[18], ResNet[15], AlexNet[19], and ChexNet[20] have many layers which take much time for training on a standard CPU. However, training a neural network with thousands of images takes more than days over a CPU, hence we trained our network on a cloud-based GPU. The hardware specifications include 12GB of NVIDIA TESLA RAM and 8 Cores of CPU. To speed up the training process, the input will be given in batches from the training set. Training the network includes three main steps. Initially, the images are loaded from the training dataset and preprocessed based on the input dimensions of the network. Here the images are resized to the aspects of 256x256x1, and the networks first layer has the same number of neurons. Second, we need to segment the pathologies affected regions of the lungs where we will be using convolutions and pooling layers. This feature extraction is achieved by training the network with input images and the masked areas from the CSV files. Lastly, the network is qualified to be a classifier. To this, labels are provided to the CXR images as vectors which contain value 1 for the pneumonia (positive class) and 0 for the non-pneumonia case. The proposed network is 69 layers convolutional architecture, taken as inspiration from u-net architecture. A sequence of layers, namely convolutions, batch normalization, and up sampling layers are arranged in the intended network for feature extraction. Leaky RelU incorporates the removal of linearity between the layers by adjusting the weights of the net- work. This architecture is trained over 25 epochs, and an epoch is defined as one complete forward pass and backward pass. The weights get updated in the forward pass and updated in the backward pass based on the given loss function. 69 layers are used to get accurate result, it gets out fitted if more than 69 layers are used, it gets under fitted if less than 69 layers are used. We use IOU BCE loss and Adam optimizer as the parameters and both, train and validation sets are obtained using the dataset generator described in the dataset section.

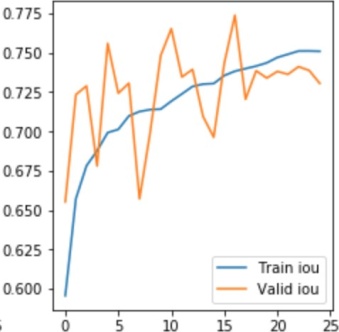
# Results and Experiments

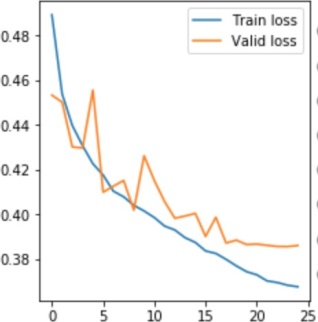
The results are evaluated based on the performance of the network on the test dataset. The test dataset is generated using the dataset generator with a batch size of 25 and images resized to 256 x 256. As discussed in the previous section, the original dataset is divided into two parts, one for training and one for validation. Out of the 26,684 images, 6012 are considered for validation purposes. The model is given a set of 3000 CXR images to segment and identify the pathologies, in this case for Pneumonia. The labeled dataset consists of 1001 images of pneumonia masked images. Main metrics used to evaluate the model are the accuracy, loss and iou for the network. Accuracy is defined as the ratio of correctly predicted pneumonia images on the dataset to the total number of pneumonia identified images. Before the training, all the weights of the network are initiated to zero mean, biases are randomly initialized in the range -1 to 1 with a standard deviation of 0.001. The train and validation losses have decreased remarkably, from 0.49 to 0.39 and 0.45 to 0.41 respectively. The model could successfully achieve 97.4% accuracy on the trained dataset, 97.2% accuracy on the validation set, and 90.2% accuracy on the test dataset. The train and validation iou increased throughout, reaching values of 0.74 and 0.72 respectively. Validation accuracy, validation loss, and validation intersection over union (iou) values are compared throughout the whole training process which is represented visually below. If pneumonia is detected, bounding boxes are drawn around the detected areas in the CXRs. Below are the results and the experiments that are obtained after the training.



**Table 1** Layers in the Proposed CNN architecture

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Layer | Type | Input Size | Output Size | No. of Filters | Filter Size |
| Layer 1 | Conv | 256x256x1 | 256x256x32 | 32 | 3x3 |
| Layer 2-5 | Batch+Leaky+Conv+MaxPool | 256x256x32 | 128x128x64 | 64 | 1x1 |
| Layer 6-17 | (Batch+Leaky+Conv)x4 | 128x128x64 | 128x128x64 | 64 3x3 |  |
| Layer 18-21 | Batch+Leaky+Conv+MaxPool | 128x128x64 | 64x64x128 | 128 | 1x1 |
| Layer 22-33 | (Batch+Leaky+Conv)x4 | 128x128x64 | 64x64x128 | 128 | 3x3 |
| Layer 34-37 | Batch+Leaky+Conv+MaxPool | 64x64x128 | 32x32x256 | 256 | 1x1 |
| Layer 38-49 | (Batch+Leaky+Conv)x4 | 32x32x256 | 32x32x256 | 256 | 3x3 |
| Layer 50-53 | Batch+Leaky+Conv+MaxPool | 32x32x256 | 16x16x512 | 512 | 1x1 |
| Layer 54-65 | (Batch+Leaky+Conv)x4 | 16x16x512 | 16x16x512 | 512 | 3x3 |
| Layer 66-68 | Batch+Leaky+Conv | 16x16x512 | 16x16x1 | 1 | 1x1 |
| Layer 69 | Upsampling | 16x16x1 | 256x256x1 | - | - |

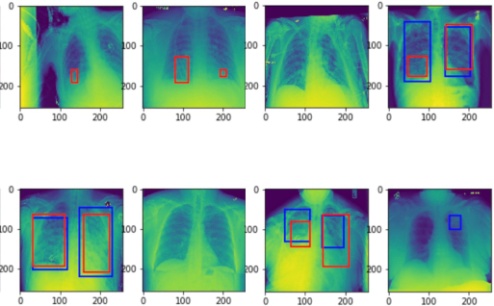




**Fig.4**:Graph depicting the train and validation loss.

**Fig.5**:Graph depicting the train and validation.

**Fig. 6**: Bounding boxes drawn for the test images obtained using the parameters from the trained model.



# Conclusion

In our paper, we have proposed a CNN architecture for the detection of pneumonia, this can be generalized to Tuberculosis. Now, instead of resorting to manual techniques for finding out the presence of lung abnormalities, we put forth an easier and reliable way to automatically detect pneumonia. This is the need of the hour which could be used in areas where people are more prone to Pneumonia/Tuberculosis. The complexity of the proposed technique is high when dealing with bigger datasets.

# References

1. Bhutia, R., Narain, K., Devi, K.R., et al. Direct and early detection of Mycobacterium tuberculosis complex and rifampicin resistance from sputum smears, Int. J. Tuberc. Lung Dis.,2013
2. Priya, E., Srinivasan, S.: "Automated object and image level classification of TB images using support vector neural network sssclassifier", Biocybernetics Biomed. Eng., 2016, 36, (4), pp. 670–678
3. B. van Ginneken, S. Katsuragawa, B. M. ter Haar Romeny, K. Doi, and M. A. Viergever, "Automatic detection of abnormalities in chest radiographs using local texture analysis," IEEE transactions on medical imaging, vol. 21, no. 2, pp. 139–149,2002
4. HorvÃa˛th G. et al. (2009) A CAD System for Screening X- rayChestRadiography.In:DÃu˝sselO.,SchlegelW.C.(eds)World Congress on Medical Physics and Biomedical Engineering, Septem- ber 7 - 12, 2009, Munich, Germany. IFMBE Proceedings, vol 25/5. Springer, Berlin, Heidelberg.
5. Guo G., Wang H., Bell D., Bi Y., Greer K. (2003) KNN Model-Based Approach in Classification. In: Meersman R., Tari Z., Schmidt D.C. (eds) On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE. OTM 2003. Lecture Notes in Computer Science, vol 2888. Springer, Berlin,Heidelberg
6. L. Hogeweg, C. Mol, P. A. de Jong, R. Dawson, H. Ayles, and B. van Ginneken, "Fusion of local and global detection sys- tems to detect tuberculosis in chest radiographs," in International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, 2010, pp.650–657
7. S. Jaeger, A. Karargyris, S. Antani and G. Thoma, "Detect- ing tuberculosis in radiographs using combined lung masks," 2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, San Diego, CA, 2012, pp. 4978- 4981
8. A. Karargyris, J. Siegelman, D. Tzortzis, S. Jaeger, S.Candemir,

Z. Xue, K. Santosh, S. Vajda, S. Antani, L. Folio et al., "Combina- tion of texture and shape features to detect pulmonary abnormalities in digital chestx-rays,"International journal of computer assisted

Radiology and surgery,vol.11,no.1,pp.99âA˘S¸106,2016.

1. P. Maduskar, R. H. Philipsen, J. Melendez, E. Scholten, D. Chanda, H. Ayles, C. I. Sa ÌA˛nchez, and B. van Ginneken, "Auto- matic detection of pleural effusion in chest radiographs," Medical image analysis, vol. 28, pp. 22–32,2016.
2. O’Shea, Keiron Nash, Ryan. (2015). An Introduction to Convolutional Neural Networks. ArXive-prints.
3. Koushik, Jayanth. (2016). Understanding Convolutional Neural Networks.
4. Srivastava, Nitish, Hinton, Geoffrey E., Krizhevsky, Alex, Sutskever,IlyaandSalakhutdinov,Ruslan."Dropout:asimpleway to prevent neural networks from overfitting.." Journal of Machine Learning Research 15 , no. 1 (2014): 1929-1958.
5. Vihar Kurama, Samhita Alla, Rohith Vishnu K, " Image Seman- tic Segmentation Using Deep Learning", International Journal of Image, Graphics and Signal Processing(IJIGSP), Vol.10, No.12, pp. 1-10, 2018.DOI:10.5815/ijigsp.2018.12.01
6. Batch Normalization: Accelerating Deep Network Training byReducing Internal Covariate Shift, Sergey Ioffe, Christian Szegedy, arXiv:1502.03167.
7. Deep Residual Learning for Image Recognition, Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun,arXiv:1512.03385
8. Searching for Activation Functions Prajit Ramachandran, Bar- ret Zoph, Quoc V. Le,arXiv:1710.05941.
9. Loshchilov, Ilya and Frank Hutter. "SGDR: Stochastic Gradient Descent with Restarts." CoRR abs/1608.03983 (2016): n.Pag.
10. A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images,"2009.
11. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet clas- sification with deep convolutional neural networks, in Advances in neural information processing systems", 2012, pp.1097–1105.
12. CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning, CheXNet: Radiologist-Level Pneu- monia Detection on Chest X-Rays with Deep Learning, Pranav Rajpurkar, Jeremy Irvin, Kaylie Zhu, Brandon Yang, Hershel Mehta, Tony Duan, Daisy Ding, Aarti Bagul, Curtis Langlotz, Katie Shpan- skaya, MatthewP. Lungren, Andrew Y. Ng,arXiv:1711.05225