Using Convolutional Neural Networks to Analyze X-Ray Radiographs for Multi-Label Classifications of Thoracic Diseases

Tiffany Zhan, Sarah Deniz, Adrian, Ng, Patricio Gonzalez, Ivy Whaley, Dennis Garcia
Sam Vinh, Jeremy Eddy, Felix Zhan, Vince Choi, Justin Zhan*
AEOP/RET Programs
University of Nevada, Las Vegas
*University of Arkansas

***Abstract*—Currently, it takes approximately 6 to 8 weeks from the initial doctor's examination to diagnose lung disease. This could potentially lead to the patient's condition worsening, the disease becoming unmanageable, or may lead to the patient's death. In order to aid doctors in the accurate and more timely diagnosis of their patients, we propose the use of convolutional neural networks for computer-aided diagnosis. Our application uses image recognition to identify the traits of various diseases in radiographs to successfully diagnose a patient. This is done through training a CNN with a dataset of 112,120 images of lung diseases. The model was tested with a resulting validation accuracy of 93 percent. The application will benefit patients suffering from these illnesses as it is time-efficient, cost-effective, and more accurate than manual diagnosis.**

***Index Terms*—convolutional neural network, deep learning, lung diseases, computer-aided diagnosis, accuracy, disease diagnosis, biomedical imaging, thoracic diseases, neural network**

I. Introduction

Lung disease is one of the leading causes of death and disability in the world [1]. For the early detection of these diseases, medical imaging, such as radiography, magnetic resonance imaging, and ultrasound scans were invented as a sub-discipline of biomedical engineering [2]. As technology becomes increasingly integrated into medicine, medical imaging has advanced to become a core component of lung disease diagnosis. With the implementation of this technology into the various processes of diagnosing diseases, image processing has been used to provide immense value in detecting diseases that may be unseen by a physician while also automating the task of diagnosis. Though incredibly vital to treatment, diagnostic images require interpretation by an expert to identify the exact cause of the symptoms. Even then, misdiagnosis is possible as more than 250,000 patients die per year due to medical error in the United States alone [3]. Other issues patients deal with in regards to medicine are the increasing demand for healthcare and the inevitable shortage of skilled professionals. It is speculated that there will be a shortage of approximately 40,000 to 120,000 physicians by the year 2032 in the United States [4]. Implementing a diagnostic system that will swiftly, effectively, and accurately diagnose patients greatly reduces the amount of time and money spent on manual diagnosis, consultations and screening methods. Various technological advancements, such as robotic surgery [5], nanotherapeutics [6], and computer-aided detection [7] have already been achieved in the field of biomedical engineering, but the potential of using deep learning for medical imaging applications has yet to be fully explored.

Our goal to facilitate the integration of machine learning into the medical field can be accomplished by constructing a convolutional neural network (CNN) to detect thoracic diseases in X-rays radiographs. Computers possess the capability to store and process large quantities of data to recognize the underlying patterns. Our program achieved a validation accuracy of 92.94 percent, which is higher than the state-of-the-art architectures [8]. When our application is implemented, patients will be able to get immediate feedback, rather than potentially waiting weeks for their results using traditional methods.

Other researchers have created technologies with a similar purpose, but there are two major drawbacks. Their models were either trained with limited datasets or yielded poor results [9]. In comparison, our model utilizes a dataset that contains more than 100,000 images while maintaining an accuracy rate exceeding 90 percent.

II. Related Works

Previous works on image analysis, using deep learning, reveal that CNNs significantly improve the accuracy of disease detection in comparison to other models. Abiyev and Ma'aitah used CNNs, backpropagation neural networks (BpNNs), and competitive neural networks (CpNNs) to diagnose various chest diseases with single label classifications [10]. The results from these researchers are as follows: in the case of the BpNN, the highest accuracy for the training set was approximately 99 percent, and it was approximately 90 percent for the validation data. The results were much lower with the CpNN: 85 percent on the training and validation data. Their claim that CNNs are

This research is supported by the Department of Defense and National Science Foundation.

978-1-7281-3783-4/20/$31.00 ©2020 IEEE

the most efficient neural network model is backed up by their results; the training data had an accuracy of 100 percent, and the validation data had an accuracy of 93 percent, which is the highest accuracy level the researchers reported. However, they achieved these results using a small sample size of 620 images, and their model lacks the ability to detect more than one disease in an X-ray radiograph.

Hattikatti also used CNNs to classify lung diseases using 30 lung CT scans and achieved an accuracy of 94 percent, compared to 87 percent accuracy when using a Support Vector Machine (SVM), demonstrating that CNNs have better accuracy rates when using scans to classify diseases [11]. SVMs exist as another viable technique for disease classification by separating the input data with a hyperplane at a distinctive position in N-dimensional space. Although the accuracy drops with an SVM, it performs less strenuous computations and demands lower processing power.

Similarly, Khobragade *et al.* made use of an artificial neural network (ANN) in order to process images of chest radiographs and identify lung diseases that would otherwise be difficult to find [12]. The proposed method took the form of four steps: image processing, segmentation, feature extraction, and image classification. Image preprocessing removed irrelevant data on the radiograph, recovering useful information, strengthening the region of interest, and simplifying its features. The segmentation was done through the use of intensity and discontinuity edge detection to find the boundaries of the lungs. Intensity converted a grayscale image into a binary image, and discontinuity detected edges in the image. Image classification occurs through the use of a feed-forward neural network.

Chang *et al.* used deep learning, specifically an ImageNet-pretrained Resnet50 model to analyze tongue fissures to make a diagnosis [9]. The issue they faced before the creation of the model was that different practitioners had different diagnoses for the same patient. Prior to Chang *et al.*, no one had created a model to use deep learning visualization methods for tongue analysis. After they tested the images through the CNN, they applied Gradient-weighted Class Activation Mapping (Grad-CAM) to constrain important image regions that contributed most to the classifications. CNNs require large quantities of data, so their dataset of 500 images is insufficient to extrapolate into real-world application. Although, the researchers reported a low accuracy rate of 70 percent, their incorporation of Grad-CAM detected and visualized most of a patient's fissures.

Wang *et al.* also utilized deep learning algorithms when they compiled an X-ray database to assist in the development of computer-aided diagnosis [13]. Researchers determined the eight common thoracic pathology keywords and subsequently employed Natural Language Processing techniques to label the data from their archives based on radiological reports attached to the images. As the patients may develop more than one disease, Wang *et al.* utilized a multi-label classification and localization framework to generate bounding boxes around the locations of thoracic ailments. Their team adapted the object localization technique to help train a Deep Convolutional Neural Network (DCNN) that would yield a multi-label classification of diseases. For their architecture, they operated ImageNet pre-trained models that resulted in varying accuracy rates for each disease, from as low as 16 percent for nodules to as high as 99 percent for cardiomegaly; hence, the detection rates are unreliable.

When referring to computer-aided diagnosis (CAD), Shiraishi *et al.* mentioned that CAD is being used at an increasing rate in the field of radiology and is commonly used in the detection of breast cancer, pulmonary embolism and prostate cancer. [14]. One downside of the method Shiraishi *et al.* details is that CAD is only used as an alternative to radiological analyses. The researchers supplemented radiologists with ANNs, similarly to Khobragade *et al.*, for data classification [12]. They measured the impact of ANNs by testing the performance of ANNs individually, radiologists without ANNs, and radiologists supported by ANNs. The subsequent evaluation with the area under the receiver operating characteristics curve (AUC) revealed that radiologists without the support of ANNs received the lowest AUC score of 0.81, ANNs alone obtained an AUC score of 0.85, and the highest AUC score of 0.87 employed the efforts of radiologists supported by ANNs. Thus, their study displayed the harmonious integration of machine learning into professional medical diagnosing; the combinations of both experts and neural networks increased classification accuracy.

Zhang *et al.* used image processing, feature extraction, and machine learning classifiers to analyze CT brain scans [15]. For image processing and feature extraction, they targeted four specific features: grayscale, shape, texture, and symmetry. Each category of features went through a Radial Basis Function Neural Network to assist in classifying whether the scan is normal or abnormal. The dataset, once again, was limited to only 212 images and achieved between 83 to 86 percent accuracy.

Alternative solutions using machine learning, particularly hidden Markov models (HMM), can also be implemented in disease detection. HMMs are statistical models containing sets of finite states. Transition probabilities dictate the relationships between states and lead to observations. HMMs have been utilized to detect cough sounds in continuous ambulatory recordings [16]. S. Matos *et al.* used 821 minutes of noise with 2473 coughs signals to train their model. The key metrics to evaluate their model performance include sensitivity, specificity, and positive predictive value; the respective median percentages are 85.7, 99.9, and 94.7. Other works include [17]–[67].

III. Methodology

For our research, we chose to build a model using convolutional neural networks. In this section we will discuss the model architecture we used to learn classifications of lung diseases. At a high level Figure 1 shows the abstraction of our model.

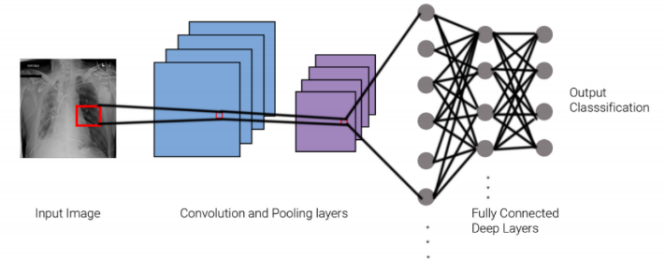

Fig. 1. The process of how images are labeled with their corresponding disease(s)

A. General Notations

Before we introduce the framework, we will briefly discuss some general notations used to explain our model.
Let X be the set of pixels $x_0, x_1, ..., x_n$ for an arbitrary image and T denote the vector containing true classifications $t_0, t_1, ..., t_m$ corresponding to the same image. The goal is for the model to estimate a function $f : X \rightarrow Z$ such that the predicted classification Z estimates the true classification. In other words, we want to minimize the follow function:

$$\varepsilon = |Z - T| \tag{1}$$

We will primarily denote an arbitrary layer number as l, weights as $w^l_{i,j}$ where i and j are the node numbers from the current layer to the next respectively.

B. Image Preprocessing

The dataset we chose to investigate, which is shown in Figure 2, consists of 112,120 images provided by [68]. These images comprise of 1024 $\times$ 1024 pixels with a 3-channel input. Inherently, this large dataset creates processing and memory issues without the use of a supercomputer. To compensate, we decided to downscale the images to resolution sizes of 256 $\times$ 256 and 160 $\times$ 160.

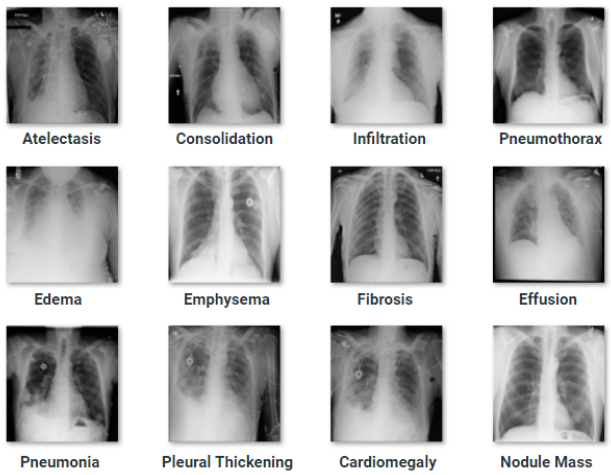

Fig. 2. National Institute of Health thorax disease dataset

Researchers have determined that rendering images at a lower resolution and grayscaling images generally does not affect the accuracy of model during training [69]. As a result, we chose to downscale our image set to make a more memory efficient model.

As a disclaimer, X-ray resolutions conventionally exceed a resolution of 3000 $\times$ 3000, which should take more computational time, but yield better accuracy results.

C. Deep CNN Model

After researching the low accuracy models generated from AlexNet [70], GoogLeNet, VGGNet-16, and ResNet-50 [13], we decided to create our own model architecture. We will briefly detail how the model works.

In general, before our model converges, our initial weights and biases are randomly generated with a normal distribution $n(\mu = 0, \sigma = 1)$.

The model starts by converting an arbitrary image into an array of pixels X. We utilize convolution screens with window sizes of 5 $\times$ 5. These convolution windows capture segments of pixels and summarizes the data in those windows. Each window can be calculated as a linear combination

$$S^{ft}_{l,m} = \sum_{n=i}^{|Window|} (x^l_n \cdot w^{ft}_{n,m}) \tag{2}$$

where f denotes the feature map number. In this case, we can treat each convolved window as a node.

For each node in the feature maps, we apply the Rectified Linear Unit (ReLU) as our activation function for faster convergence [71].

$$Y^{ft}_{l,m} = ReLU(S^{ft}_{l,m} + b_{l,m}) \tag{3}$$

$Y^{ft}_{l,m}$ becomes the new input node for the proceeding layer, with $b_{l,m}$ denoting the appended bias term we add to prevent overfitting.

Following each convolution layer, we pool from the layer of different window sizes. Pooling allows our model to reduce the dimensionality by summarizing the windows even further using MaxPooling, which takes the maximum pixel value to represent the window.

After three pairs convolution and pooling layers, we start a fully connected deep neural network. Each node in the feature map of the pooling layer has an associated weight to reach any node in the first hidden layer. Furthermore, to reduce overfitting, we add an additional dropout condition, to reduce the amount of active nodes in each layer.

Because our dataset contains multilabel classifications, where each disease is independent of each other, we utilized the Sigmoid function at the end to preform independent binary classifications [72] [73].

$$z_m = Sigmoid(S_{l,m}) \tag{4}$$

In our dataset, we use one-hot encoding to transfer a series of multilabel classification strings into binary truth vectors T.

Once we had resized the images into smaller dimensions (256 $\times$ 256 and 160 $\times$ 160), we first decided to use a one-hot encoder in order to encode the labels for the data into a one hot array. This allows the categories of the diseases to go under the process of binarization so that they are able to be used as a feature to train the model [74]. After obtaining our estimated classification, we can calculate the loss between our targeted output and the expected output [75]. Specifically, we utilize the mean square error.

$$E = \frac{1}{2}(Z - T)^2 \tag{5}$$

Using this error, we perform a backpropagation process to update and adjust our weights and biases. Altogether, these variables, Var can be adjusted in the following way:

$$Var_{n+1} = Var_n - \eta \times (UpdateRule) \tag{6}$$

where the corresponding update rule is dependent on the optimizer and η denotes the learning rate. For our model, we experimented with both the adam optimizer and the traditional stochastic gradient descent method.

D. OneHotEnconding and Multilabel Classification

From the available dataset, a corresponding multilabel classification is assigned to each image, representing the patient's different illnesses. Within the class of diseases, $D = [d_0, d_1, ..., d_{14}]$ a patient with multiple diseases will have a truth-vector T_n = [Pneumonia, Fibrosis, Mass,...]. The representation we chose to express these multilabel classifications is to convert it into a binary vector with 1 representing the presence of a disease and 0 indicating the absence of it. The example truth-vector T_n can be encoded, using OneHotEnconding with a multilabel binarizer, and can be represented as $T_n = [1, 0, 0, 0, ..., 1, 0]$.

IV. Experiments

A. Dataset

The National Institutes of Health provided the dataset used in this experiment [68]. The dataset consists of 112,120 X-ray images from 30,805 unique subjects. Each image has been normalized to a size of 1024 by 1024 pixels. The X-ray scans possess a corresponding disease label located in a CSV file. The original authors employed Natural Language Processing (NLP) to attach these labels by data-mining the disease classifications from the associated radiological reports. Due to NLP extraction, the labeling accuracy is estimated to be above ninety-percent but the data could still contain incorrect labels.

Figure 2 demonstrates an example of the pairings of disease labels in the training set with their corresponding thorax X-ray scan. The 14 diseases labels identified within this dataset include Atelectasis, Consolidation, Infiltration, Pneumothorax, Edema, Emphysema, Fibrosis, Effusion, Pneumonia, Pleural thickening, Cardiomegaly, Nodule, Mass, and Hernia.

B. Major Libraries used for CNN model implementation

For the construction of our CNN model, we coded using Python 3.7 as it supports an extensive number of machine learning and data manipulation libraries. The most-used libraries of our model are:

1) *Keras*: A high level deep neural networks library written in Python.
2) *Tensorflow*: An open-sourced symbolic math library employed as a backend for machine learning applications.
3) *Pandas*: A Python data analysis and manipulation library.
4) *Pillow*: A Python imaging library to interact with various image file formats.

Minor libraries include: numpy, scikit learn, glob, matplotlib, and h5py.

C. Implementation

As mentioned in III-C, we used OneHotEncoding. One error we came across was that only using OneHotEncoding changed the dimensionality of the arrays to be a shape different from our desired input shape. We fixed this problem by using MultiLabelBinarizer, a transformer which one-hot encodes data with multiple labels while still retaining the desired input shape. [76]

We utilized the glob library in order to loop through the different images in the dataset and append them into an array. However, loading all of 112,120 images into the array at once produced a memory error.

There were several adjustments that were made in an attempt to accommodate the lack of memory and optimize the accuracy. One solution was to load the images by batch, appending the images and labels to the arrays one at a time until 8,000 images are loaded in the array and training the model with those images. 2,000 images and labels are then appended to a different array for validation. Afterwards, both the training and validation arrays are cleared and the process continues until all 112,120 images are used. The only resolution with which we were able to use all 112,120 images in the dataset without loading by batch was 160 $\times$ 160. Besides 160 $\times$ 160, we also used 256 $\times$ 256, with 50,000 images.

Our final model comprised of three convolutional layers with 4, 8, and 32 feature maps, respectively; the feature maps' corresponding filter size went from (5 $\times$ 5), (3 $\times$ 3), and (3 $\times$ 3). Each convolutional layer was followed by a max-pooling layer of (2 $\times$ 2) receptive fields to downsample the images. Then dropout layers were laid between each pair of convolutional and max-pooling to reduce overfitting [77]. We flattened the model before adding three dense fully connected layers.

V. Results

Table I shows the results from our model, and Table II shows the results from our model compared to other models' validation results.

TABLE I
RESULTS FROM OUR MODEL

Image Resolution	Accuracy		
	Number of Images	*Accuracy (%)*	*Optimizer*
*256x256	50000	94.0	adam
*256x256	50000	93.0	SGD
*160x160	112120	93.5	adam
*160x160	112120	92.1	SGD
**256x256	50000	92.0	adam
**256x256	50000	91.9	SGD
**160x160	112120	92.9	adam
**160x160	112120	92.0	SGD

*Training. **Validation.

For the batch that used images with a resolution of 256 × 256, Figures 3 and 4 show the difference between the training and validation accuracy. The 256 × 256 batch with the optimizer changed from adam to SGD is shown in Figures 5 and 6. The accuracy and loss for the set consisting of images with resolution of 160 × 160 is displayed in Figures 7 and 8. The optimizer for the 160 × 160 batch was adjusted to SGD, which is represented in Figures 9 and 10.

As can be seen in Table II, when the results we achieved are compared to models such as AlexNet, GoogLeNet, and ResNet-50, we have a significantly higher accuracy. The results from AlexNet had an accuracy of 63 percent for their validation data, and they used 1.2 million images with 256 × 256 resolution [70]. GoogLeNet used the same database with a lower resolution and their accuracy was 64 percent [78]. ResNet-50 had the same resolution but fewer images than GoogLeNet and those researchers achieved an accuracy of 69 percent [79].

TABLE II
COMPARISON OF VALIDATION RESULTS

Image Resolution	Accuracy		
	Number of Images	*Accuracy (%)*	*Source*
256x256	1200000	63.4	AlexNet
224x224	1200000	63.5	GoogLeNet
224x224	50000	69.3	ResNet-50
256x256	50000	92.9	Our Model

One additional factor that influenced our results was the number of epochs with which we trained the model. Initially, we only ran the model for 5 epochs for time-efficiency. For more in-depth results, the optimized model was run for 100 epochs. As shown in Figure 3, the accuracy oscillates due to the micro-adjustments of the weights in model, but increased from around 91 to 93 percent overall, which means that it is essential that it is run for 100 epochs to be able to calculate the true accuracy.

VI. DISCUSSION AND FUTURE WORKS

The highest accuracy we were able to achieve for our validation dataset was 92.94 percent. Although this is not a foolproof method of analysis for hospital usage, it can

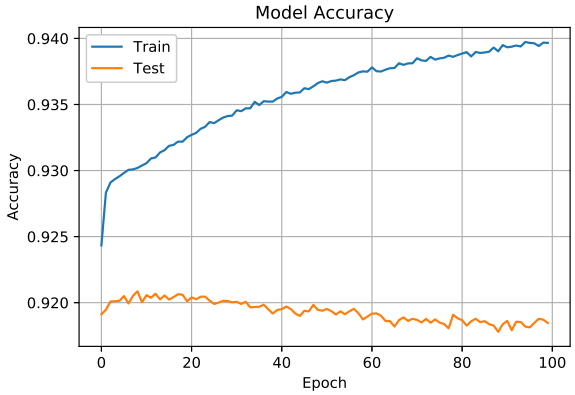

Fig. 3. Accuracy for the 256 × 256 batch with adam as the optimizer

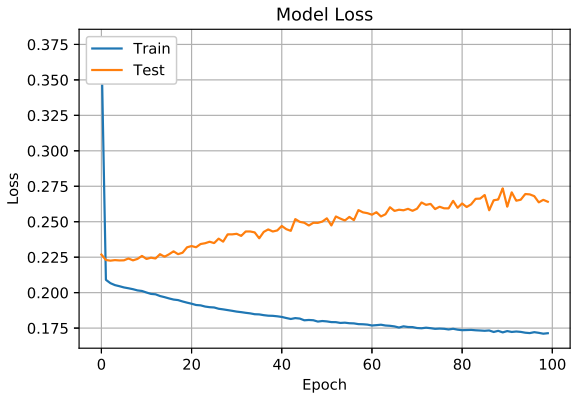

Fig. 4. Loss value for the 256 × 256 batch with adam as the optimizer

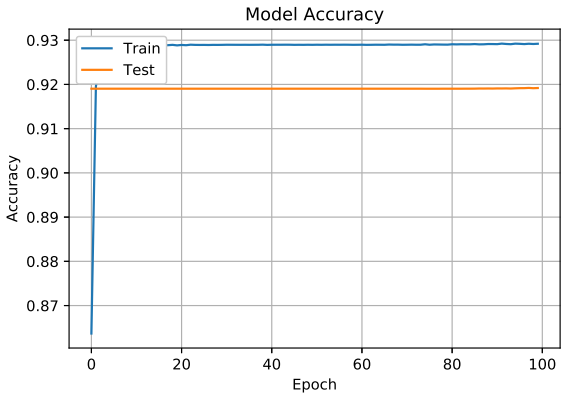

Fig. 5. Loss value for the 256 × 256 batch with SGD as the optimizer

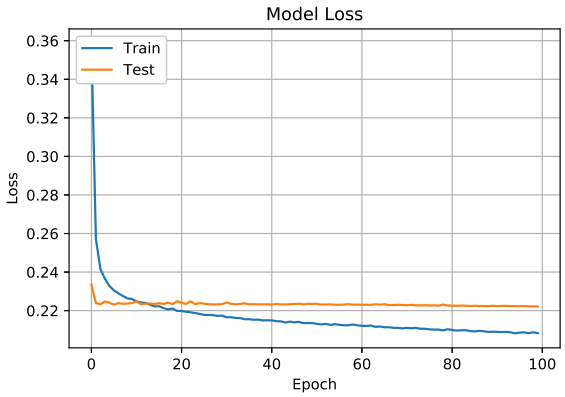

Fig. 6. Loss value for the 256 × 256 batch with SGD as the optimizer

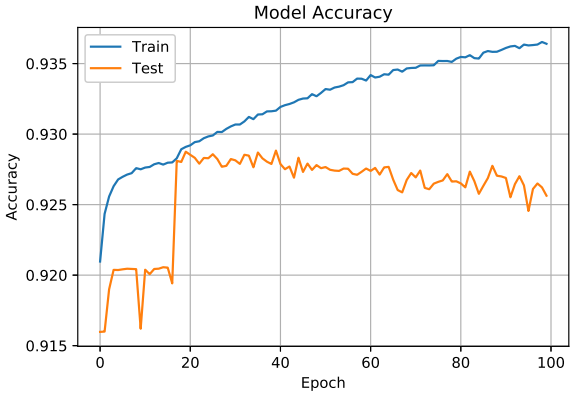

Fig. 7. Accuracy for the 160 × 160 batch with adam as the optimizer

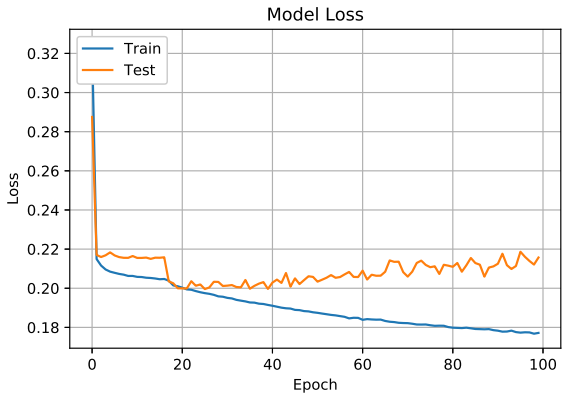

Fig. 8. Loss value for the 160 × 160 batch with adam as the optimizer

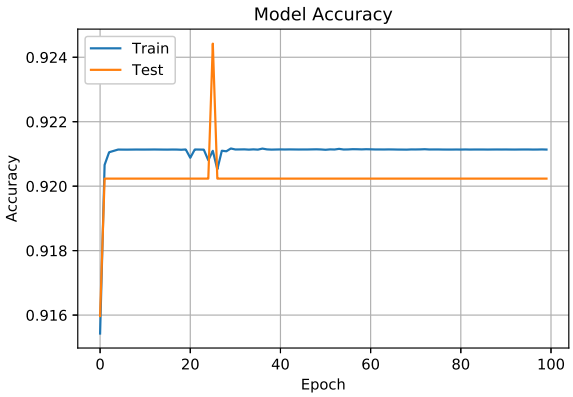

Fig. 9. Accuracy for the 160 × 160 batch with SGD as the optimizer

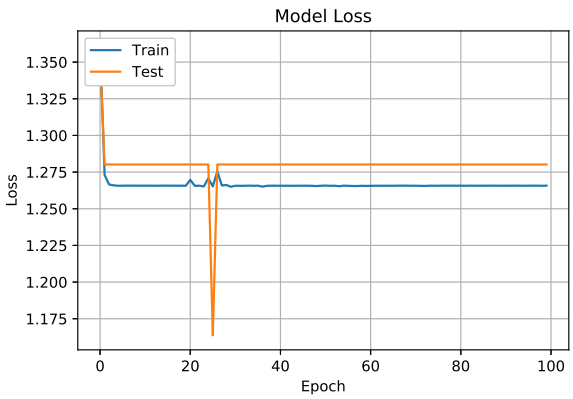

Fig. 10. Loss for the 160 × 160 batch with SGD as the optimizer

be used to supplement a radiologist's diagnosis. There was no significant difference between using adam and the SGD as the optimizer in the validation results, but the training results suggest that using adam produces a higher accuracy percentage. This also applies to how the resolution and number of images impacted the accuracy; in the training, higher pixel resolution led to higher training accuracy but had a minuscule impact on the validation set.

In the field of medicine, it is imperative that diagnoses be made quickly and accurately; after all, a person's life hangs in the balance. By achieving 93 percent accuracy with multi-label classifications, our model exhibits substantial improvement compared to previous research, but it can be further developed.

The next step for this project is to create a user-friendly computer application. This will make it more accessible to the public, as well as ensure that radiologists are able to use it in hospitals. The model foundation can be extended to incorporate more diseases as long as a larger dataset is compiled. We hope to be able to shorten the amount of time it takes to diagnose patients so doctors are able to administer the appropriate treatment as soon as possible.

References

[1] F. of International Respiratory Societies, "The global impact of respiratory disease – second edition," 2017.

[2] "Medical image fusion: A survey of the state of the art," Jan 2014.

[3] "Study suggests medical errors now third leading cause of death in the u.s. - 05/03/2016," May 2016.

[4] I. M. Ltd, "The complexities of physician supply and demand: Projections from 2017 to 2032," 2019.

[5] V. Vitiello, S. Lee, T. P. Cundy, and G. Yang, "Emerging robotic platforms for minimally invasive surgery," *IEEE Reviews in Biomedical Engineering*, vol. 6, pp. 111–126, 2013.

[6] O. Kosheleva, T. Lai, Y. Chan, M. Hsiao, S. Ng, C. Chen, and N. G. Chen, "Investigation of nanoparticle-assisted ultrasound therapy for treating implanted breast tumors in mice," in *2017 IEEE 17th International Conference on Nanotechnology (IEEE-NANO)*, July 2017, pp. 672–674.

[7] H. R. Roth, L. Lu, J. Liu, J. Yao, A. Seff, K. Cherry, L. Kim, and R. M. Summers, "Improving computer-aided detection using convolutional neural networks and random view aggregation," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1170–1181, May 2016.

[8] C. S. Lee, P. G. Nagy, S. J. Weaver, and D. E. Newman-Toker, "Cognitive and system factors contributing to diagnostic errors in radiology : American journal of roentgenology : Vol. 201, no. 3 (ajr)." [Online]. Available: https://www.ajronline.org/doi/full/10.2214/AJR.12.10375

[9] W. Chang, H. Chu, and H. Chang, "Tongue fissure visualization with deep learning," in *2018 Conference on Technologies and Applications of Artificial Intelligence (TAAI)*, Nov 2018, pp. 14–17.

[10] R. H. Abiyev and M. K. S. Ma'aitah, "Deep convolutional neural networks for chest diseases detection," Aug 2018.

[11] P. Hattikatti, "Texture based interstitial lung disease detection using convolutional neural network," in *2017 International Conference on Big Data, IoT and Data Science (BID)*, Dec 2017, pp. 18–22.

[12] S. Khobragade, A. Tiwari, C. Y. Patil, and V. Narke, "Automatic detection of major lung diseases using chest radiographs and classification by feed-forward artificial neural network," in *2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES)*, July 2016, pp. 1–5.

[13] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers, "Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 3462–3471.

[14] J. Shiraishi, Q. Li, D. Appelbaum, and K. Doi, "Computer-aided diagnosis and artificial intelligence in clinical imaging," *Seminars in Nuclear Medicine*, vol. 41, no. 6, pp. 449 – 462, 2011, image Perception in Nuclear Medicine. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0001299811000742

[15] W. Zhang and X. Wang, "Feature extraction and classification for human brain ct images," pp. 1155–1159, Aug 2007.

[16] S. Matos, S. S. Birring, I. D. Pavord, and D. H. Evans, "An automated system for 24-h monitoring of cough frequency: The leicester cough monitor," *IEEE Transactions on Biomedical Engineering*, vol. 54, no. 8, pp. 1472–1479, Aug 2007.

[17] M. Schwob, J. Zhan, and A. Dempsey, "Modeling cell communication with time-dependent signaling hypergraphs," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2019.

[18] S. Chobe and J. Zhan, "Advancing community detection using keyword attribute search," *Journal of Big Data*, vol. 6, no. 83, 2019.

[19] C. Chiu and J. Zhan, "An evolutionary approach to compact dag neural network optimization," *IEEE Access*, vol. 7, no. 1, pp. 178 331 – 178 341, 2019.

[20] ——, "Deep learning for link prediction in dynamic networks using weak estimators," *IEEE Access*, vol. 6, no. 1, pp. 35 937 – 35 945, 2018.

[21] M. Bhaduri and J. Zhan, "Using empirical recurrences rates ratio for time series data similarity," *IEEE Access*, vol. 6, no. 1, pp. 30 855–30 864, 2018.

[22] J. Wu, J. Zhan, and S. Chobe, "Mining association rules for low frequency itemsets," *PLOS ONE*, vol. 13, no. 7, 2018.

[23] P. Ezatpoor, J. Zhan, J. Wu, and C. Chiu, "Finding top-k dominance on incomplete big data using mapreduce framework," *IEEE Access*, vol. 6, no. 1, pp. 7872–7887, 2018.

[24] P. Chopade and J. Zhan, "Towards a framework for community detection in large networks using game-theoretic modeling," *IEEE Transactions on Big Data*, vol. 5, no. 1, pp. 27 354–27 365, 2017.

[25] M. Bhaduri, J. Zhan, and C. Chiu, "A weak estimator for dynamic systems," *IEEE Access*, vol. 5, no. 1, pp. 27 354–27 365, 2017.

[26] M. Bhaduri, J. Zhan, C. Chiu, and F. Zhan, "A novel online and non-parametric approach for drift detection in big data," *IEEE Access*, vol. 5, no. 1, pp. 15 883–15 892, 2017.

[27] C. Chiu, J. Zhan, and F. Zhan, "Uncovering suspicious activity from partially paired and incomplete multimodal data," *IEEE Access*, vol. 5, no. 1, pp. 13 689 – 13 698, 2017.

[28] R. Ahn and J. Zhan, "Using proxies for node immunization identification on large graphs," *IEEE Access*, vol. 5, no. 1, pp. 13 046–13 053, 2017.

[29] J. Zhan and B. Dahal, "Using deep learning for short text understanding," *Journal of Big Data*, vol. 4, no. 34, pp. 1–15, 2017.

[30] J. Zhan, S. Gurung, and S. P. K. Parsa, "Identification of top-k nodes in large networks using katz centrality," *Journal of Big Data*, vol. 4, no. 16, 2017.

[31] J. Zhan, T. Rafalski, G. Stashkevich, and E. Verenich, "Vaccination allocation in large dynamic networks," *Journal of Big Data*, vol. 4, no. 2, pp. 161–172, 2017.

[32] J. Zhan, V. Gudibande, and S. P. K. Parsa, "Idenfication of top-k influential communities in large networks," *Journal of Big Data*, vol. 3, no. 16, 2016.

[33] H. Selim and J. Zhan, "Towards shortest path identification on large networks," *Journal of Big Data*, vol. 3, no. 10, 2016.

[34] X. Fang and J. Zhan, "Sentiment analysis using product review data," *Journal of Big Data*, vol. 2, no. 5, pp. 1–14, 2015.

[35] P. Chopade and J. Zhan, "Structural and functional analytics for community detection in large-scale complex networks," *Journal of Big Data*, vol. 2, no. 1, pp. 1–28, 2015.

[36] J. Zhan and X. Fang, "A computational framework for detecting malicious actors in communities," *International Journal of Privacy, Security, and Integrity*, vol. 2, no. 1, pp. 1–20, 2014.

[37] A. Doyal and J. Zhan, "Towards ddos defense and traceback," *International Journal of Privacy, Security, and Integrity*, vol. 1, no. 4, pp. 299–311, 2013.

[38] J. Zhan, J. Oommen, and J. Crisostomo, "Anomaly detection in dynamic systems using weak estimator," *ACM Transaction on Internet Technology*, vol. 11, no. 1, pp. 53–69, 2011.

[39] J. Zhan and X. Fang, "Social computing: The state of the art," *International Journal of Social Computing and Cyber-Physical Systems*, vol. 1, no. 1, pp. 1–12, 2011.

[40] N. Mead, M. S., and J. Zhan, "Integrating privacy requirements considerations into a security requirements engineering method and tool," *International Journal of Information Privacy, Security and Integrity*, vol. 1, no. 1, pp. 106–126, 2011.

[41] J. Zhan, "Granular computing in privacy-preserving data mining," *International Journal of Granular Computing, Rough Sets and Intelligent Systems*, vol. 1, no. 3, pp. 272–288, 2010.

[42] J. Wang, J. Zhang, and J. Zhan, "Towards real-time performance of data privacy protection," *International Journal of Granular Computing, Rough Sets and Intelligent Systems*, vol. 1, no. 4, pp. 329–342, 2010.

[43] J. Zhan, "Secure collaborative social networks," *IEEE Transaction on Systems, Man, and Cybernetics, Part C*, vol. 40, no. 6, pp. 682–689, 2010.

[44] J. Zhan, H. C., I. Wang, T. Hsu, C. Liau, and W. D., "Privacy-preserving collaborative recommender systems," *IEEE Transaction on Systems, Man, and Cybernetics, Part C*, vol. 40, no. 4, pp. 472–476, 2010.

[45] H. Park, J. Hong, J. Park, J. Zhan, and D. Lee, "Attribute-based access control using combined authentication technologies," *IEEE Transaction on Mobile Computing*, vol. 9, no. 6, pp. 824–837, 2010.

[46] I. Wang, C. Shen, J. Zhan, T. Hsu, C. Liau, and D. Wang, "Empirical evaluations of secure scalar product," *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, vol. 39, no. 4, pp. 440–447, 2009.

[47] N. Mead, V. Viswanathan, and J. Zhan, "Incorporating security requirements engineering into standard lifecycle processes," *International Journal of Security and Its Applications*, vol. 2, no. 4, pp. 67–80, 2008.

[48] J. Zhan, L. Chang, and S. Matwin, "Privacy-preserving multi-party decision tree induction," *International Journal of Business Intelligence and Data Mining*, vol. 2, no. 2, pp. 197–212, 2007.

[49] ——, "Building k-nearest neighbor classifiers on vertically partitioned private data," *International Journal of Network Security*, vol. 1, no. 1, pp. 46–51, 2005.

[50] J. Zhan and S. Matwin, "Privacy preserving support vector machine classification," *International Journal of Intelligent Information and Database Systems*, vol. 1, no. 3/4, pp. 356–385, 2005.

[51] F. Zhan, A. Martinez, N. Rai, R. McConnell, M. Swan, M. Bhaduri, J. Zhan, L. Gewali, and P. Oh, "Beyond cumulative sum charting in non-stationarity detection and estimation," *IEEE Access*, 2019.

[52] A. Hart, B. Smith, S. Smith, E. Sales, J. Hernandez-Camargo, Y. M. Garcia, F. Zhan, L. Griswold, B. Dunkelberger, M. R. Schwob, S. Chaudhry, J. Zhan, L. Gewali, and P. Oh, "Resolving intravoxel white matter structures in the human brain using regularized regression and clustering," *Journal of Big Data*, vol. 6, 2019.

[53] F. Zhan, "Hand gesture recognition with convolution neural networks," in *Proceedings of IEEE 20th International Conference on Information Reuse and Integration for Data Science*, Los Angeles, CA, USA, July 31-August 1 2019.

[54] ——, "How to optimize social network influence," in *Proceedings of the IEEE Second International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*, Cagliari, Sardinia, Italy, June 3-5 2019.

[55] E. Aguilar, J. Dancel, D. Mamaud, D. Pirosch, F. Tavacoli, F. Zhan, R. Pearce, M. Novack, H. Keehu, B. Lowe, J. Zhan, L. Gewali, and P. Oh, "Highly parallel seedless random number generation from arbitrary thread schedule reconstruction," in *IEEE International Conference on Big Knowledge*, Beijing, China, November 10-11 2019.

[56] E. Hunt, R. Janamsetty, C. Kinares, C. Koh, A. Sanchez, F. Zhan, M. Ozdemir, S. Waseem, O. Yolcu, B. Dahal, J. Zhan, L. Gewali, and P. Oh, "Machine learning models for paraphrase identification and its applications on plagiarism detection," in *IEEE International Conference on Big Knowledge*, Beijing, China, November 10-11 2019.

[57] M. Bhaduri, J. Zhan, C. Chiu, and F. Zhan, "A novel online and non-parametric approach for drift detection in big data," *IEEE Access*, vol. 5, pp. 15 883–15 892, 2017.

[58] C. Chiu, J. Zhan, and F. Zhan, "Uncovering suspicious activity from partially paired and incomplete multimodal data," *IEEE Access*, vol. 5, pp. 13 689–13 698, 2017.

[59] D. Pintado, V. Sanchez, E. Adarve, M. Mata, Z. Gogebakan, B. Cabuk, C. Chiu, J. Zhan, L. Gewali, and P. Oh, "Deep learning based shopping assistant for the visually impaired," in *IEEE International Conference on Consumer Electrics*, Las Vegas, USA, January 2019.

[60] F. Fessahaye, L. Perez, T. Zhan, R. Zhang, C. Fossier, R. Markarian, C. Chiu, J. Zhan, L. Gewali, and P. Oh, "T-recsys: A novel music recommendation system using deep learning," in *IEEE International Conference on Consumer Electrics*, Las Vegas, USA, January 2019.

[61] F. Zhan, G. Laines, S. Deniz, S. Paliskara, I. Ochoa, I. Guerra, M. Pirouz, C. Chiu, S. Tayeb, E. Ploutz, J. Zhan, L. Gewali, and P. Oh, "An efficient alternative to personalized page rank for friend recommendations," in *IEEE Consumer Communications and Networking Conference*, Las Vegas, USA, January 2018.

[62] F. Zhan, G. Laines, S. Deniz, S. Paliskara, I. Ochoa, I. Guerra, S. Tayeb, C. Chiu, M. Pirouz, E. Ploutz, L. G. Justin Zhan and, and P. Oh, "Prediction of online social networks users' behaviors with a game theoretic approach," in *IEEE Consumer Communications and Networking Conference*, Las Vegas, USA, January 2018.

[63] S. Tayeb, M. Pirouz, B. Cozzens, R. Huang, M. Jay, K. Khembunjong, S. Paliskara, F. Zhan, M. Zhang, J. Zhan, and S. Latifi, "Toward data quality analytics in signature verification using a convolutional neural network," in *IEEE International Conference on Big Data*, Boston, USA, December 2017, pp. 2644–2651.

[64] P. Chopade and J. Zhan, "Large-scale big data networks analytics and community detection," in *IEEE International Symposium on Technologies for Homeland Security*, Waltham, MA USA, April 2017.

[65] ——, "Efficient detection of communities and prediction of abnormal events, situational awareness in large complex networks," in *IEEE International Symposium on Technologies for Homeland Security*, Waltham, MA USA, April 2017.

[66] P. Chopade, J. Zhan, and M. Bikdash, "Micro-community detection and vulnerability identification for large critical networks," in *IEEE International Symposium on Technologies for Homeland Security*, Waltham, Massachusetts, USA, May 2016.

[67] ——, "Node attributes and edge structure for large-scale big data network analytics and community detection," in *IEEE International Symposium on Technologies for Homeland Security*, Boston, USA, April 2015.

[68] "National institutes of health chest x-ray dataset." [Online]. Available: https://nihcc.app.box.com/v/ChestXray-NIHCC

[69] R. Wu, S. Yan, Y. Shan, Q. Dang, and G. Sun, "Deep image: Scaling up image recognition."

[70] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks." [Online]. Available: https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[71] I. Dabbura and I. Dabbura, "Coding neural network - forward propagation and backpropagtion," Apr 2018. [Online]. Available: https://towardsdatascience.com/coding-neural-network-forward-propagation-and-backpropagtion-ccf8cf369f76

[72] S. Sharma, "Activation functions in neural networks," Sep 2017. [Online]. Available: https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6

[73] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, *Dive into Deep Learning*. University of California, Berkeley, 2019, http://www.d2l.ai.

[74] M. Cassel and F. Lima, "Evaluating one-hot encoding finite state machines for seu reliability in sram-based fpgas," in *12th IEEE International On-Line Testing Symposium (IOLTS'06)*, July 2006, pp. 6 pp.–.

[75] R. Varma, "Picking loss functions - a comparison between mse, cross entropy, and hinge loss." [Online]. Available: https://rohanvarma.me/Loss-Functions/

[76] K. Chuahan, U. Surakod, V. B. Dayanand, K. Basavaraju, and S. Radhakrishna, "Image based product recommendation system."

[77] Y. Gal and Z. Ghahramani, "Bayesian convolutional neural networks with bernoulli approximate variational inference," Jan 2016.

[78] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions." [Online]. Available: https://www.cs.unc.edu/ wliu/papers/GoogLeNet.pdf

[79] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition." [Online]. Available: https://arxiv.org/pdf/1512.03385.pdf