Autoencoder for Bioinformatics Using Convolutional Neural Networks

Karl Dill kad210004@utdallas.edu
Michael Muruthi mkm150430@utdallas.edu
Farishah Nahrin fxn170230@utdallas.edu
Richard Padilla rmp200004@utdallas.edu
Jibin Prince jpc140330@utdallas.edu
Jignesh Satam jxs210042@utdallas.edu
Jacob Villegas jacob@utdallas.edu

***Abstract*—**According to the international peer-reviewed journal, BMJ Quality & Safety, approximately "twelve million adults in the United States are misdiagnosed every year" [3]. Such errors have heightened the pressures, for both medical professional and researchers, to utilize methods that can help reduce inaccuracies during medical diagnoses. Many medical diagnoses are discovered through medical imaging, such as X-rays, which are one of the most used diagnostic tools today. The analysis of medical imaging and the creation of the X-ray was first created "accidently" by Wilhelm Conrad Roentgen in 1895. After only one year of this discovery, physicians immediately began to use this as an official medical diagnostic tool to examine human organs and bones. The rapid progression and evolution of radiology and medical diagnoses, in healthcare, has created an impact on the way technologists strive to find new ways to uncover and detect diagnoses from medical images or videos. The diagnoses that are detected by machine learning algorithms versus humans are still being studied and compared, till this today. Thus, in the past century, the discovery of diseases through images, have greatly evolved, as the usage of neural networks and artificial intelligence are being applied to consume large records of data, to deduce diagnoses. In return, the breakthroughs in machine learning and big data have helped improved the accuracy, proficiency, and reliability of diagnoses through medical images.

Keywords*—*autoencoder, one-hot encoder, deep neural networks, convolutional neural networks, artificial intelligence, big data, machine learning, biotechnology, bioinformatics, biology, pneumonia, medical diagnosis, diseases, medical imaging, medical image analysis, medicine

I. Introduction

When medical imaging was first discovered, physicians, radiologists, and medical researchers were primarily the ones who were responsible for the diagnostics of medical images. However, the evolution of technology, machine learning and artificial intelligence has left some of the medical diagnosis practices, on the shoulders of technologists. The purpose of using technology and machine learning algorithms during medical diagnoses, is to simply help physicians and radiologists, in their work, and create devices that also make use of these algorithms. Currently, there are two major devices that physicians and radiologists use to detect diagnoses on images, which are the "CADx, Computer Aided Detection and the CAD, Computer Aided Diagnosis" [4]. The accuracy of these devices is incredibly crucial since the algorithm used in these devices will ultimately result in how "treatments and procedures are determined" [4]. According to S. M. Anwar, et al., it is important to have "high evaluation metrics, such as recall, F-measure, precision, during a medical image analysis, since the quality of these numbers determines if the algorithm used on the medical imaging devices are highly accurate or inaccurate" [4].

According to D. R. Sarvamangala, et al., "convolutional neural networks have outperformed humans, when it comes to understanding and diagnosing diseases from medical images" [5]. The

convolutional neural network is not the only technique that is used to determine medical diagnoses from images. In recent years, the devices that are used to comprehend images use a vast variety of machine learning techniques such as, "clustering, decision tree learning, k-means nearest neighbor, random forests, and Boltzmann machines" [5]. The purpose of using machine learning techniques to understand images, is to use a large set of data to "discover and extract the desirable features and use these features to determine a diagnosis, when an image is passed through a device that contains this algorithm" [5].

In order for a system or device, which uses "machine learning techniques, to be considered valuable during medical image diagnoses," the algorithm has to have several key descriptors, which make it usable to the medical community [6]. One of the requirements include having good performance, which means that the algorithm used on the medical image analysis has to "extract important information from the large set of data, and "the diagnosis has to be accurate, when a new use case, or image, is passed through the algorithm" [6]. The second requirement is to see how well the machine learning algorithm can "deal with missing pieces of information from the dataset" [6]. It is very common for "patient data and records to not hold all the information needed for determining a diagnosis, therefore it is essential for the algorithm to take care of data that is not complete" [6]. The next technique that is required is to deal with meaningless, noisy data. Since many "medical records of data can contain unnecessary information, the machine learning algorithm needs to find a way to endure this" [6]. And lastly, the one the other techniques that is helpful for a machine learning algorithm to have is to "reduce the number of tests" [6]. According to I. Kononenko, it is "costly and time insensitive for the patient data to be collected frequently, thus it is best to have an algorithm that is able to recognize a diagnosis from a fewer amount of information in a dataset" [6]. One way to ensure that the machine learning algorithm is able to be accurate with a smaller amount of data, is to "determine the correct subset of data, which contains a suitable range of features, and pass it through the algorithm" [6].

Machine learning algorithms, specifically neural networks, can be used to decipher a variety of different medical illnesses and diseases, therefore our paper aims to focus on X-ray images of the human lungs and chest only. Thus, the objective of this paper is to illustrate how the convolutional neural network, using the autoencoder framework, can be used to accurately classify and train X-ray images of lungs, so that it can better understand the medical image. Following an examination of our dataset and pre-processing, this paper will discuss the model creation and techniques used, then this paper will conclude by stating the results and future work.

II. Background Work

A. Our Chosen Dataset

To formulate our analysis of the bioinformatic medical diagnoses detection, our team has determined to use the Pneumonia Chest X-Ray images from Kaggle. The Kaggle Chest X-ray dataset contained two separate directories: train, and test. And "within these two directories, it contained two different categories: pneumonia and normal" [2]. The

"Normal" directory contains images of chests that do not have pneumonia. Whereas the "Pneumonia" directory contains images of chests that contain either bacterial or viral pneumonia. The Normal directory contained 1,341 in the training set and 234 in the test set. The Pneumonia directory contained 3,875 images in the training set and 390 in the test set. Also, all of the images are in greyscale. These chest X-ray images were derived from "groups of children from one to five years old from the Guangzhou Women and Children's Medical Center in China" [2]. The images were already "assessed for

quality and the dataset had removed low-quality pictures of the X-ray" [2]. In terms of reliability of this dataset, the pictures were "scored by two specialist physicians, and in order to account for scoring errors, the dataset was checked by an additional expert" [2].

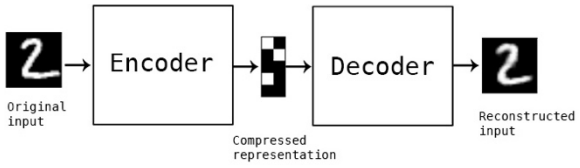
The reason why our team has chosen to utilize this dataset is because pneumonia is a disease that has been determined as the "one of the leading cause of deaths in children" [1]. "Bacterial and viral infectious agents are the leading causes for pneumonia in children, and X-rays are the sole diagnostic tool to deduce if pneumonia is present in the lungs" [1].

B. Pre-Processing and Feature Engineering

Now that the Chest X-Ray dataset has been illustrated, the pre-processing of the dataset was determined through a certain number of steps. Our first step in pre-processing, was to prepare the images, so that it can properly pass through the Convolutional Neural Network. We loaded all of our training, test, and validation datasets in a public and readable Amazon S3 bucket. Then, we created a table, within our Databricks notebook, which displays the file path, modification time, length of image, the image .jpeg itself, the label, and the features. We created a label column that labels the image as "1," if the image originated from the Pneumonia directory, and it also labels the image as "0," if the image originated from the Normal directory. As mentioned previously, we added a features column, which converts the binary image into an array. Also due to a Databricks error, two hundred eighty images were not being converted into a byte image array. Therefore, we have chosen to remove those two hundred eighty sets of images. After the removal of these images, the error was resolved. As mentioned earlier, since these images were already cross-checked and graded by expert physicians, we chose to not remove any more images, beside the specific two hundred eighty images. Also, since Kaggle already separated the dataset into Train and Test directories, we did not need to use a library to split the dataset into training and test sets. And lastly, for pre-processing, we resized the image to [28, 28], divided it by 255, and vectorized it from -1 to 1, on the intensity. And we decided to use a batch size of 128, in order to output optimal results.

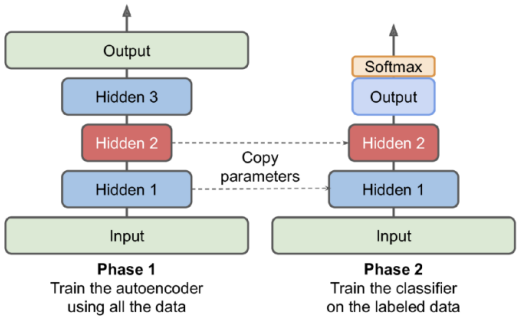
III. Theoretical/Conceptual Study of the Algorithm **and** How We Applied This Algorithm

Before deep diving into the way our team applied the Convolution Neural Network, with the autoencoder framework, we will explain the algorithm we have used, first. First, we used the autoencoder model with Convolutional Neural Network for classifying the X-ray images as Normal or Pneumonia. First, we "created a convolutional base that takes the shape of the imported image tensors, which will output each image as a 2D tensor of the shape" [10]. The autoencoder contains "two dense layers that convert the image, which are known as the encoder and decoder" [11]. The autoencoder will essentially "train the input image to be duplicated to the output, and it will encode the image into a lower dimensional vector" [11]. Next, the decoder will take that "lower dimensional vector and convert, or remanufacture it, back to an image" [11]. The autoencoder we build is "compressing the data, while not ruining the format of the image" [11].



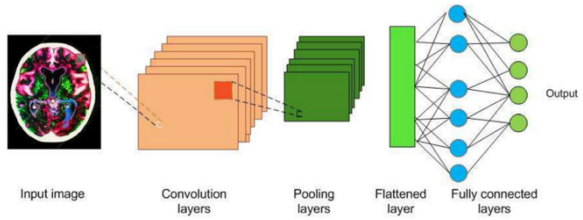
Then, we automated the labeling of each image, by creating a transformer to handle that. And as mentioned in the Pre-Processing section, we converted the images into a byte size array using NumPy; we used a transformer to automate this execution, for each image as well. We identified to have 4,931 features, after converting each image into a vector array. To train the model, we used a batch size of one hundred twenty-eight, we used ten epochs, and then used the train dataset, and ran the autoencoder on the train, since the train dataset

can generate itself using itself. Next, we saved these weights for the classification autoencoder. Next, we one-hot encoded our data which "transforms the categorical data into a binary output and creates a number of individual categories into 0s and 1s" [12]. We created a SoftMax dense layer for the classification of the one-hot encoded labels. A SoftMax layer is used "as a mathematical equation, which will transform the vector of these array of numbers into an array of probabilities" [13]. It is essentially acting as an "activation function in the convolutional neural network. Using the encoder from the autoencoder function, we combined it with the SoftMax layer to detect these One-Hot encoded labels. Then, we pass the weights of the image encoder onto the new encoder. We transferred the weights of the original encoder to another one, so that it can learn, and in a sense, get trained, as show in the figure below [14].



Next, we compiled the autoencoder with the weights to determine the categorical accuracy for the two one-hot encoded labels (normal and pneumonia). Then, we compile the autoencoder with weights, which provided us with an iteration series of the statistics from the fitted classification model.

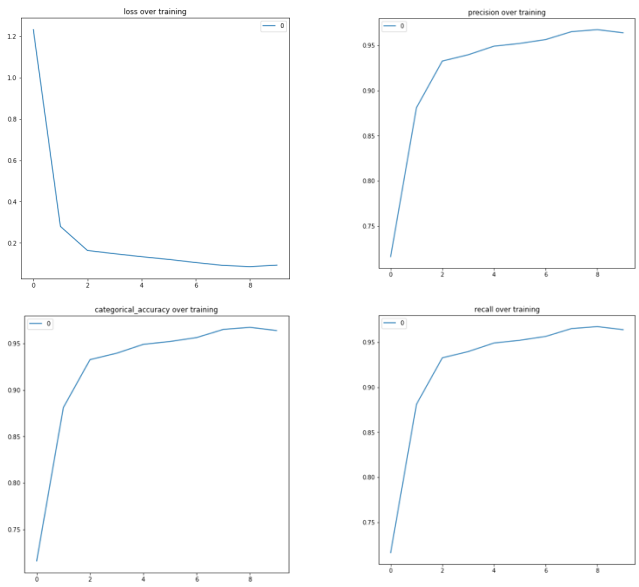
The convolutional neural network had "it's first major advancement, when it was used to detect cancer, on an accuracy of 89%, with GoogleNet" [5]. On the other hand, humans were only able to accurately determine the diagnosis with a "70% accuracy rate" [5]. Therefore, the "convolutional neural network contains convolutional filters, which learns the features from the images automatically, and extracts the features, so that it can be understood for medical image diagnoses" [5]. The architecture of the convolutional neural network is shown below and "comprises of pooling and fully connected layers before the output can contain the learned traits of the image" [5].



IV. Results and Analysis

As a result, we utilized the autoencoder framework and the convolutional neural network to train our dataset. Using this technique, our algorithm "processed the images, recognized the images, learned the features needed from the input data, and then extracted the important features" [2]. We have built this algorithm that can further process and identify images that are normal (healthy) versus pneumonic in a very accurate manner. When we trained the model, we noticed that the value of our loss, precision, accuracy, and recall improved, at each iteration of the epoch, as shown in the figures below.

We have created iteration series graphs, which display the results of our evaluation metrics over the training dataset. Our Databricks notebook contains more figures and statistics, in addition to these below.



As a result, we determined that our test loss is 6%, the accuracy is 97.5%, the false positive and negative value is 123, the Precision and Recall are approximately 97.5% and the AUC value is 99.4%. Therefore, we have determined that our algorithm is as nearly correct since our accuracy is high. And our loss is low since our model is highly confident in its predictions. Loss is a means of determining how far off from the correct predicted value, that our prediction is. And accuracy is the ratio of correct predictions to the incorrect predictions, and is not absent in underlying bias. Therefore, our algorithm can be used to predict the images nearly-accurately, as either healthy or as pneumonic.

V. Conclusion and Future Work

We utilized convolutional neural networks and the autoencoder framework technique to train our dataset, and as a result, we were able to see optimal performances. By applying this model to our dataset, we have gained an insightful observation on the impacts of the convolutional neural network on image processing. In the future, this exercise could be further altered to better take advantage of other features of the python language, such as Horovod or Elephas. In addition, this algorithm can be further used and implemented, in medical devices used for medical diagnoses; for example, a physician can use a device or site, which implements this algorithm, to insert X-ray images. Then the device can, in return, conclude if the lungs, in the X-Ray image, is more likely to be healthy and normal, or contain pneumonia.

References

[1] D. S. Kermany, *et al.,* "Identifying medical diagnoses and treatable diseases by image-based Deep Learning," Cell, 22-Feb-2018. [Online].Available: https://www.cell.com/cell/fulltext/S0092-8674(18)30154-5. [Accessed: 28-Nov-2022].

[2] P. Mooney, "Chest X-ray images (pneumonia)," Kaggle, 24-Mar-2018. [Online]. Available: https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia. [Accessed: 25-Nov-2022].

[3] H. Singh, A. N. Meyer, and E. J. Thomas, "The frequency of diagnostic errors in outpatient care: Estimations from three large observational studies involving us adult populations," BMJ quality & safety, 17-Apr-2014. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/24742777/. [Accessed: 27-Nov-2022].

[4] S. M. Anwar, M. Majid, A. Qayyum, M. Awais, M. Alnowami, and M. K. Khan, "Medical image analysis using Convolutional Neural Networks: A Review - Journal of Medical Systems," SpringerLink, 08-Oct-2018. [Online]. Available: https://link.springer.com/article/10.1007/s10916-018-1088-1. [Accessed: 27-Nov-2022].

[5] D. R. Sarvamangala and R. V. Kulkarni, "Convolutional neural networks in medical image understanding: A survey - evolutionary intelligence," SpringerLink, 03-Jan-2021. [Online]. Available: https://link.springer.com/article/10.1007/s12065-020-00540-3. [Accessed: 27-Nov-2022].

[6] I. Kononenko, "Machine learning for medical diagnosis: History, state of the art and perspective," Artificial Intelligence in Medicine, 16-Jul-2001. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S093336570100077X. [Accessed: 29-Nov-2022].

[7] M. Chumbita, C. Cillóniz, P. Puerta-Alcalde, E. Moreno-García, G. Sanjuan, N. Garcia-Pouton, A. Soriano, A. Torres, and C. Garcia-Vidal, "Can artificial intelligence improve the management of pneumonia," MDPI, 17-Jan-2020. [Online]. Available: https://www.mdpi.com/2077-0383/9/1/248. [Accessed: 28-Nov-2022].

[8] IBM Cloud Education, "What are convolutional neural networks?," IBM, 20-Oct-

2020. [Online]. Available: https://www.ibm.com/cloud/learn/convolutional-neural-networks. [Accessed: 29-Nov-2022].

[9] N. BM, “What is an encoder decoder model?,” Medium, 02-May-2022. [Online]. Available: https://towardsdatascience.com/what-is-an-encoder-decoder-model-86b3d57c5e1a. [Accessed: 30-Nov-2022].

[10] “Convolutional Neural Network (CNN) Tensorflow Core,” TensorFlow. [Online]. Available: https://www.tensorflow.org/tutorials/images/cnn. [Accessed: 30-Nov-2022].

[11] “Intro to autoencoders :   Tensorflow Core,” TensorFlow. [Online]. Available: https://www.tensorflow.org/tutorials/generative/autoencoder. [Accessed: 30-Nov-2022].

[12] N. Hespe, “Building autoencoders on sparse, one hot encoded data,” Medium, 28-Sep-2020. [Online]. Available: https://towardsdatascience.com/building-autoencoders-on-sparse-one-hot-encoded-data-53eefdfdbcc7. [Accessed: 30-Nov-2022].

[13] J. Brownlee, “Softmax activation function with python,” MachineLearningMastery.com, 23-Jun-2020. [Online]. Available: https://machinelearningmastery.com/softmax-activation-function-with-python/. [Accessed: 30-Nov-2022].

[14] A. Géron, “Hands-on machine learning with scikit-learn, Keras, and Tensorflow, 2nd edition,” O'Reilly Online Learning. [Online]. Available: https://www.oreilly.com/library/view/hands-on-machine-learning/9781492032632/. [Accessed: 30-Nov-2022].