DEVELOPING AI ALGORITHMS TO CLASSIFY PATHOLOGIES IN CHEST X-RAY IMAGES

An Undergraduate Research Scholars Thesis

by

RODA AL-SUBAIE, OUMAIMA BOUHALI, LOLWA AL-NASR, AND YARA ELGAZAR

Submitted to the LAUNCH: Undergraduate Research office at
Texas A&M University
in partial fulfillment of requirements for the designation as an

UNDERGRADUATE RESEARCH SCHOLAR

Approved by
Faculty Research Advisor: Dr. Jim Ji

May 2023

Major: Electrical Engineering

RESEARCH COMPLIANCE CERTIFICATION

Research activities involving the use of human subjects, vertebrate animals, and/or biohazards must be reviewed and approved by the appropriate Texas A&M University regulatory research committee (i.e., IRB, IACUC, IBC) before the activity can commence. This requirement applies to activities conducted at Texas A&M and to activities conducted at non-Texas A&M facilities or institutions. In both cases, students are responsible for working with the relevant Texas A&M research compliance program to ensure and document that all Texas A&M compliance obligations are met before the study begins.

We, Roda Al-Subaie, Oumaima Bouhali, Lolwa Al-Nasr, and Yara Elgazar, certify that all research compliance requirements related to this Undergraduate Research Scholars thesis have been addressed with our Faculty Research Advisor prior to the collection of any data used in this final thesis submission.

This project did not require approval from the Texas A&M University Research Compliance & Biosafety office.

TABLE OF CONTENTS

ABSTRACT

Developing AI Algorithms to Classify Pathologies in Chest X-Ray Images

Roda Al-Subaie, Oumaima Bouhali, Lolwa Al-Nasr, and Yara Elgazar
Department of Electrical and Computer Engineering
Texas A&M University

Faculty Research Advisor: Dr. Jim Ji
Department of Electrical and Computer Engineering
Texas A&M University

Radiologists are in charge of detecting and diagnosing diseases by means of x-rays, Magnetic Resonance Imaging (MRIs), Computed Tomography (CT) scans, and other medical imaging techniques. Of these imaging techniques, chest x-rays are widely popular, as they can detect a various number of diseases related to the heart and lungs. In Qatar, the majority of diseases detected by x-rays include COVID-19, Pneumonia, Tuberculosis (among immigrant workers), and lung cancer. While chest x-rays are really helpful in detecting these diseases, one of the biggest problems concerning the field of radiology is that radiologists often have trouble diagnosing the patient, even though they can detect that there is something wrong. As a result, they often have to repeat the x-ray, consult other doctors, or resort to other medical imaging techniques. Consequently, a lot of time is wasted and costs are amounted, meaning there is inefficiency in the system. Furthermore, exposing the patient to repeated scans is risky. An inefficient radiologist can lead to unsatisfied patients and a prolonged treatment plan. This can be dangerous, especially when the patient's disease is high risk and requires an immediate response.

In the recent years, radiologists have begun to adopt Artificial Intelligence (AI) to help them resolve these inefficiencies by aiding them in the diagnosis of diseases. The purpose of this project is to create an AI algorithm that will help the radiologist to diagnose a patient based on a given image of a chest x-ray. The algorithm will be accessed through a graphical user interface (GUI), where the radiologist can input an image and get the diagnosis as an output. There are essentially two subsystems nested into one another: the AI algorithm and the GUI. The entire system shall be called Che-X-Ray.

ACKNOWLEDGEMENTS

Contributors

We would like to thank our faculty advisor, Dr. Jim Ji, for his guidance and support throughout the course of this research.

Thanks also go to our friends and colleagues and the department faculty and staff for making our time at Texas A&M University a great experience.

Finally, thanks to our family for their encouragement and most importantly our parents for their patience and love.

All other work conducted for the thesis was completed by the students independently.

Funding Sources

No funding was used for the work conducted for the thesis.

NOMENCLATURE

AI	Artificial Intelligence
GUI	Graphical User Interface
CNN	Convolutional Neural Network

1. INTRODUCTION

1.1 Overview of Artificial Intelligence in Radiology

Chest radiography has long been a prominent field in medicine, allowing for the diagnosis of a wide range of diseases or illnesses associated with the chest. More specifically, chest x-rays have recently become one of the best technologies that radiologists use to detect these diseases, such as Pneumonia, Tuberculosis, COVID-19, and lung cancer.

With the increasing presence of AI in the medical realm, the field of radiology began to adopt certain applications of AI, particularly neural networks that detect and classify pathologies based on an x-ray image of the chest.

1.2 Motivation

One of the biggest problems concerning the field of radiology is that radiologists often have trouble diagnosing the patient, even though they can detect that there is something wrong. A lot of the times, many chest-related diseases, such as Pneumonia and Tuberculosis, are present at the same time, which makes it harder to identify the specific pathologies present in the x-ray. As a result, they often have to repeat the x-ray, consult other doctors, or resort to other medical imaging techniques. Consequently, a lot of time is wasted and costs are amounted, meaning there is inefficiency in the system. An inefficient radiologist can lead to unsatisfied patients and a prolonged treatment plan. This can be dangerous, especially when the patient's disease is high risk and requires an immediate response. Furthermore, exposing the patient to repeated scans is not very safe and poses a lot of other risks.

In the recent years, radiologists have begun to adopt Artificial Intelligence (AI) to help them resolve these inefficiencies by aiding them in the diagnosis of diseases. As such, the purpose of this project is to introduce an AI algorithm that will serve as an aid for these radiologists, as they diagnose their patients.

1.3 Purpose and Scope of Project

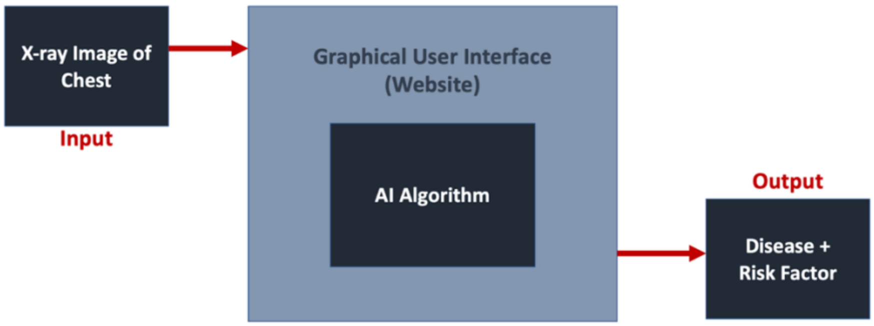

Figure 1: Overview of project.

The purpose of this project is to create an AI algorithm that will help the radiologist to diagnose a patient based on a given image of a chest x-ray. The algorithm will be accessed through a graphical user interface (GUI), where the radiologist can input an image and get the diagnosis as an output. There are essentially two subsystems nested into one another: the AI algorithm and the GUI.

The system proposed is called Che-X-Ray, which is essentially made up of two sub-systems: an AI algorithm and a GUI. These two components work hand in hand to help the radiologist come up with a diagnosis, given an image of a chest x-ray.

Figure 1 briefly illustrates how the proposed system will function. After conducting a chest x-ray, the radiologist will have an x-ray image, which they will input into the GUI. The GUI contains the AI algorithm, through which the x-ray image will go through. The algorithm will give an output that classifies the x-ray image, indicating all the present diseases in the patient. The GUI will present that output to the radiologist, clearly specifying which diseases are present and the risk factor associated with each of these diseases. The risk factor is essentially a score indicating the percent confidence that the disease is present.

1.4 Literature Review

The AI algorithms that will we will be exploring include supervised and unsupervised learning techniques, both of which have proved to have extensive applications in image classifications. In order to survey the state of knowledge in these learning techniques, we conducted a literature review on studies that have used these techniques for medical image classification. In one study, the Self-Organizing Feature Maps (SOFM) algorithm was used in order to detect the patients with COVID-19 from a chest x-ray scan. It was able to classify the images as either healthy or sick accordingly [1]. Another study using a supervised machine learning technique, specifically a deep learning technique, known as Xg-Boost (XGB). It was used to classify TB at early stages and was found effective to radiologists and to patients because of early detection, since it was a challenge for them to detect it before the condition becomes life threatening [2]. Furthermore, another study compares different types of algorithms using both supervised and unsupervised learning techniques, such as the Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Random Forest (RF), Naïve Bayes algorithm, and Decision Tree. The chest x-ray images were used to detect COVID-19 patients and the most effective method used with an accuracy of 96% was that of SVM following it with an accuracy of 92% were KNN and RF [3]. In addition to that, another study shows the effectiveness of two algorithms among other algorithms known as Logistic Regression (LR - Supervised learning technique) and Multilayer Perceptron (MLP - Unsupervised learning technique) used in detecting pneumonia in patients with an accuracy of 95.63% and 95.39% respectively [4]. It is important to note that all of these studies employed public datasets that can be found online.

1.5 Evaluation of Project

1.5.1 Quantitative Assessment of Project

Upon conducting an extensive literature review and proposing the main concepts for the proposed system, it is clear that the proposed system offers great improvements to the existing system.

Firstly, it is clearly evident in the literature review that most studies employ public, labeled datasets to train their AI algorithms. While this is great in theory, the model they create cannot be used by any radiologist. It is important to note that the dataset used greatly influences the bias of the algorithm. For example, the type of machine used to generate the x-ray images or the diversity of the people that the images are based on, all have an impact on the performance of the algorithm. Since the current systems employ public datasets, this means that the algorithms they create cannot be applied to any given image of an x-ray. They can only be tested on images of similar nature (images that have been generated by the same machine, on the same people, and from the same hospital). In this project, the proposed idea is to use local, private data, in order to be able to apply the algorithm to the local people of Qatar. Since the model would be trained on x-ray images of local people and generated by the machines provided in local hospitals, this means that the model can generalize to new test data that is specific to Qatari hospitals. This is an improvement from the existing system, since they only employ general, public data, which means that the models they create cannot generalize on specific types of new data and wouldn't be able to be applied to any specific hospital. As such, the proposed system acknowledges the bias of AI algorithms and employs appropriate datasets to cater to local needs.

Furthermore, the literature review reveals that most machine learning techniques employed are supervised learning techniques. These techniques are only able to be applied to labeled data. The proposed system suggests that we use convolutional neural networks, which have proven to yield accurate and precise results in image classification and detection problems, particularly in the field of medicine.

Finally, the added GUI component is a great improvement to the existing system, since it allows radiologists to directly use the algorithm with easy access. Currently, based on the literature review, many of these algorithms have not been made available to radiologists. If they have been made available, they are usually complicated to access or not even deployed. Therefore, our proposed system will apply a web interface, so that it can be easily be accessed to the radiologist. Instead of dealing with the algorithm directly, the radiologists are only prompted to simply upload an image and click on a button to obtain the output, which is essentially the diseases associated with the image and the probabilistic score indicating the confidence of the classification.

1.5.2 Limitations

Most of the limitations lie in the dataset used for the training of the AI algorithm. The first limitation is the range of diseases that the model is trained on. In the case of the public dataset, there are 14 different diseases that the model would be aware of. On the other hand, in the case of the private dataset, there would be much less diseases that the model would be trained to classify. As a result, the model would be specific to a specific range of diseases, meaning it wouldn't know how to classify diseases outside of this range. However, this limitation is not that much of a concern. The pathologies that the algorithms are able to classify are narrowed down the to the 14 most prevalent ones in chest x-rays. With this in mind, an expert radiologist can opt to use this tool when he is certain that the patient is most likely suffering from one of the common pathologies.

1.5.3 Risk Assessment

The ideal plan is to use a private local dataset. Since the private dataset is unlabeled, we will employ unsupervised AI algorithms. One risk would be not receiving the private dataset on time, which would mean that the ideal plan will be abandoned and use a public dataset with labeled photos. In this situation, supervised learning techniques will be used, since they work better for labeled data. There will be some modifications, while switching between two distinct types of algorithms. Furthermore, another risk is associated with the bias of the AI algorithm. Since the model is trained on certain demographic, then the model might not be able to make generalizations on tourists or visitors of the country that might need a chest x-ray.

1.5.4 Standards

Since the project deals with human data, particularly images pertaining to humans, it is important to abide by the law of Texas A&M University and Qatar when dealing with human data. Some important standards to follow and keep in mind during the collection of the dataset are maintaining patient confidentiality when dealing with medical records. Furthermore, the patient's anonymity should also be protected, when using the dataset provided by the hospital. Finally, in the creation of the GUI, the privacy of the patients must be considered. The GUI should protect the patients' privacy, by preserving their anonymity and confidentiality.

1.5.5 User Classes and Other Involved Personnel

The proposed system involves various users, however, the most important are the radiologists. More specifically, the radiologists in Qatar. Since the radiologists are only going to receive the GUI, this should be the most important component to cater to their needs. For example, it should be easy to use and navigate, so that radiologists can easily upload patients' data and receive a clear output. While patients are not the users of this proposed system, they are still involved in the process. More specifically, the patients to keep in mind are local patients of Qatar, since the dataset is most likely going to be private local data of the people living in Qatar. Even more so, the chest x-rays will be conducted under one type of machine that is used in Qatar. All of these details means that the AI algorithm is being trained on x-rays of the people of Qatar generated by machines specific to Qatari hospitals. As a result, this means that the targeted patients are people living in Qatar.

1.6 Summary of Impacts

The implementation of the proposed system may have wide ranging impacts on the hospital, physicians, and patients. The sub-sections below identify potential operational impacts and organizational impacts that will be taken into consideration as the system is developed. It will help radiologists save time and money. It will also assist to improve the classification system's efficiency.

1.6.1 Operational Impacts

Since the AI algorithm and GUI component have not yet been developed, it is hard to understand the operational impacts. However, what is certain is that the proposed system will aid radiologists in determining what kind of chest disease a patient has. Some operational impacts are outlined below.

- Time and money will be saved, as the radiologist relies on the AI instead of: repeating the x-ray, consulting other doctors, or resorting to other medical imaging techniques [15].
- More efficient system, as radiologists come to quick decisions with the help of the AI.
- Patients will be more confident in their diagnosis being accurate, as it comes from the combined judgment of a well trained doctor and a well trained algorithm.

1.6.2 Organizational Impacts

There will be some practices and policies that have to be changed when working with such a system. When the system is developed and implemented, it would be clear what the organizational impacts are. However, some possible impacts are given below.

- Privacy of patients when data and photos are examined by the system.
- The effect on the hospital’s funding, as they might need to spend money on obtaining GUIs.
- The possible need for extra training of employees to run the system: GUI and computer system.

1.7 Essential Concepts to Proposed System

- Radiology: Radiology is a medical specialty that uses imaging technologies to diagnose and treat disorders.
- Pathology: Pathology refers to a medical condition that doctors diagnose.
- Artificial Intelligence: Artificial intelligence is a simulation of human modeling capabilities that use different devices, and computer systems. AI is used in the field of health for improving doctors and patient’s outcomes, as well as reducing the cost required for clinical diagnosis and treatments. Different approaches are currently followed by various AI systems in healthcare.
- Neural Network/ Model: Neural networks are subsets of artificial intelligence that consist of various node layers, which include an input layer, an output layer, and one or more hidden layers. They are trained on a certain dataset and then tested on new inputs that they haven’t been trained on. Neural networks have been used in the field of health care for several years.

- Machine Learning: Machine learning (ML) is a branch of artificial intelligence (AI) that enables software programs to grow increasingly effective at predicting outcomes without explicitly programming them to do so. Machine learning algorithms estimate new output values using past data as input.
- Binary Cross Entropy: can be defined as a model measure that records a model's inaccurate labeling of a data class, compensating the model if variations in probability happen when categorizing the labels.
- Supervised Learning: It is a branch of machine learning that is mainly characterized by its use of labeled training and testing datasets, which are then classified accordingly.
- Unsupervised Learning: It is also another branch of machine learning that is mainly characterized by its ability to recognize patterns in data and employ clustering or grouping methods to predict the final outcome.
- Class: In neural networks used for classification, class refers to the category that the network could potentially assign an input to.
- CNN: A neural network for algorithms for deep learning that is primarily utilized for image recognition and pixel data processing applications. There are different forms of neural networks in deep learning, but CNNs are the network design of choice for identifying things.
- Binary Accuracy: A metric that determines the percentage of anticipated values that match actual values. It computes how frequently predictions match binary labels.

2. METHODS

2.1 Dataset Collection

AI algorithms are trained and tested on large datasets. In the case of this project, the purpose is to train the algorithm to be able to classify a given chest x-ray image based on 14 different pathologies. As such, the dataset is comprised of images of chest x-rays.

The dataset that was used to train the AI algorithm was comprised of two types of datasets: a public dataset and a private dataset. The public dataset consists of chest x-ray images that we obtained from a public dataset. On the other hand, the private dataset are chest x-ray images that we obtained from local hospitals in Qatar. Both datasets are labelled, meaning that each image is associated with a label indicating all of the present diseases.

The public dataset was primarily used to train our models. On the other hand, the private dataset was used both in training and testing our model. In using both types of datasets in the training, it was ensured that the model was being exposed to a diverse dataset, so that it could be more accurate in its classification. The ultimate goal was to ensure that the model is able to generalize well enough to be able to make accurate predictions on the private dataset, which is why the testing was mainly done on the private dataset.

2.1.1 Public Dataset

The public dataset was obtained from “CheXpert”, which is a large public dataset that consists of 224,316 chest radiographs of 65,240 patients [1]. Each image is gray-scaled and has a dimension of 390 by 320 pixels. These images were collected from Stanford Hospital and the X-rays were performed between October 2002 and July 2017 [1]. Each image is labeled with a 14-element vector, with each element corresponding to one of fourteen different pathologies [1]. For each pathology, a label of 1 is assigned if its presence is positive, 0 if it is negative, and U if it is uncertain. Figure 2 clearly displays the number and percentage of each image in the dataset based on their label (positive, uncertain, or negative) and based on their pathology.

Pathology	Positive (%)	Uncertain (%)	Negative (%)
No Finding	16627 (8.86)	0 (0.0)	171014 (91.14)
Enlarged Cardiom.	9020 (4.81)	10148 (5.41)	168473 (89.78)
Cardiomegaly	23002 (12.26)	6597 (3.52)	158042 (84.23)
Lung Lesion	6856 (3.65)	1071 (0.57)	179714 (95.78)
Lung Opacity	92669 (49.39)	4341 (2.31)	90631 (48.3)
Edema	48905 (26.06)	11571 (6.17)	127165 (67.77)
Consolidation	12730 (6.78)	23976 (12.78)	150935 (80.44)
Pneumonia	4576 (2.44)	15658 (8.34)	167407 (89.22)
Atelectasis	29333 (15.63)	29377 (15.66)	128931 (68.71)
Pneumothorax	17313 (9.23)	2663 (1.42)	167665 (89.35)
Pleural Effusion	75696 (40.34)	9419 (5.02)	102526 (54.64)
Pleural Other	2441 (1.3)	1771 (0.94)	183429 (97.76)
Fracture	7270 (3.87)	484 (0.26)	179887 (95.87)
Support Devices	105831 (56.4)	898 (0.48)	80912 (43.12)

Figure 2: Table of pathologies and labels of the CheXpert dataset [1].

2.1.2 Private Dataset

On the other hand, the private dataset should be obtained from a local hospital in Qatar, called the Primary Health Care Corporation (PHCC). The prevalent diseases in Qatar are Pneumonia, TB, lung cancer, and COVID-19, which means that the dataset is mainly comprised of chest x-rays of these diseases. Upon meeting with a radiologist from the PHCC, it was made clear to us that 2499 x-ray images would be provided to us, if we are able to obtain an IRB from our university, Texas A&M University (TAMU). Each image had a multi-label of 13 different categories: COVID19 findings, right upper lung zone, right middle lung zone, right lower lung zone, left upper lung zone, left middle lung zone, left lower lung zone, ground glass opacity, consolidation, reticular interstitial thickening, nodular, pleural effusion, and pneumothorax. For each class, a label of positive (1), negative (0), or not applicable (9) is assigned. It is important to note that this private dataset was not collected for this research, but they existed at a local hospital for other research. Thus, the IRB for collecting the dataset was obtained by the hospital. In order to be able to use the hospital's data, an IRB from TAMU had to be filed and submitted.

2.2 Implementation of AI Algorithm

The main component of the proposed system is the AI algorithm. The purpose of the AI algorithm is to classify a given image of a chest x-ray, based on 14 different types of possible diseases. There may be several diseases present at once in a given image.

The algorithms used are mainly types of convolutional neural networks (CNNs), since they are widely used for image classification problems. The three types of CNNs used are: DenseNet, ResNet, and MobileNet. Each of these models are characterized by the types of layers they use and the number of parameters they have.

For each type of AI algorithm, a similar method of implementation was used. Figure 3 illustrates the flow diagram of the AI algorithm. Firstly, the dataset went through image pre-processing techniques, which consists of cropping all the images to a certain size and filtering the image. After that, the dataset was split into two groups: a training set and a testing set. The training images includes all of the public dataset and 10% of the private dataset. On the other hand, the testing images includes 90% of the private dataset. Once split, the training set goes through the AI algorithm. Once the algorithm is fully trained, it becomes a trained model that the testing images can be inputted into. Ultimately, the trained model will output the pathologies present in a given image and a risk factor for each pathology, which the percent confidence that the particular pathology is present in the image.

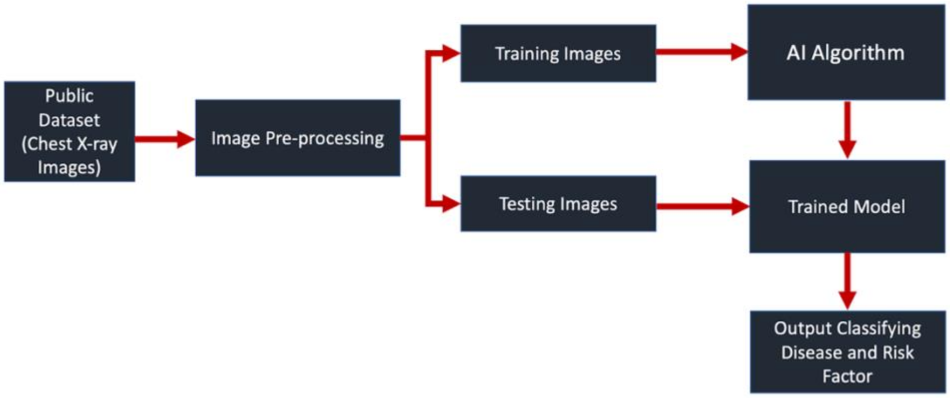

Figure 3: Flow diagram of AI Algorithm.

The entire system depicted in Figure 3 was implemented using the programming language, Python, with important machine learning libraries, such as Tensorflow.

2.2.1 Training

In terms of training the algorithm, each of the three types (MobileNet, ResNet, and DenseNet) were trained using two different techniques: training from scratch and training with transfer learning. Training from scratch implies that the model is built from scratch and the model parameters are randomly intialized. On the other hand, training using transfer learning implies that the model has been previously trained on a similar dataset, meaning that the model parameters are set at specific values. In the case of this project, the models trained with the transfer learning technique have been previously trained on the ImageNet dataset.

Algorithm 1, 2, 3: Training From Scratch

Input: epochs, learning_rate, batch_size, and train_generator.

Output: loss, metrics, and history.

```
input_tensor= keras.Input(train_generator, shape=(320, 320, 1))
*hidden layers*
y = keras.layers.GlobalAveragePooling2D()(x)
y = keras.layers.Dense(14, activation='sigmoid')(y)
model = keras.Model(input_tensor, y)
model.compile(tf.keras.optimizers.Adam(learning_rate=learning_rate),
loss=keras.losses.BinaryCrossentropy(),
```

```
metrics=[keras.metrics.BinaryAccuracy()])
8 history = model.fit(train_generator, verbose=1, epochs=epochs, batch_size= 512,
shuffle=True)
```

Algorithms 1, 2, and 3 correspond to training the DenseNet, ResNet, and MobileNet models from scratch. They clearly follow a similar structure in terms of the input being the tensor of training images (train_generator) and other parameters, such as the learning rate (learning_rate), batch size (batch_size), and number of epochs. The train_generator tensor goes into the model's input layer, which is defined in line 1. Then, each algorithm is characterized by its own set of hidden layers, which is represented in line 2. These hidden layers are any combination and number of convolutional layers, depth-wise convolutional layers, dense layers, max-pooling layers, and average pooling layers. This combination depends on whether the model being implemented is DenseNet, ResNet, or MobileNet. After that, all models end with an average pooling layer, a dense layer, and finally a 14-neuron output layer. Each neuron in the output layer has a sigmoidal activation function. Then, the model is compiled in line 6, which is where the hyperparameters and loss function are defined. Finally, the model is trained using the model.fit function in line 8.

On the other hand, algorithms 4, 5, and 6 employ transfer learning. Like algorithms 1, 2, and 3, the inputs are the hyperparameters (epochs and learning rate) and the input tensor (train_generator). For each of these algorithms, a similar Python code was used. Firstly, in line 1, the pretrained model is called, and the parameters are defined (the weights of the model, the output layer of the model, and the shape of the input layer). Then, in line 2, the first layer of the model is defined, which is a convolutional layer. This layer is then followed by the loaded and pretrained model, which is added in line 3. After the layers of the pretrained model are built, a maxpooling layer is added in line 4, followed by a dense layer in line 5. Finally, in line 6, the entire model is defined and named "model." Once the model was built, it had to be compiled, as shown in line 7. In the compilation, the learning rate, loss, and metrics of the training process are defined. Finally, in line 8, a function is used to run the training of the model. The output of the code is the training accuracy (binary accuracy) and loss (binary cross entropy) at each epoch.

Algorithm 4: MobileNet with Transfer Learning

Input: epochs, learning rate, batch_size, and train_generator.
Output: loss, metrics, and history.

```
1  mobilenet = keras.applications.MobileNetV2(include_top=False,
   weights='imagenet',input_shape=(320, 320, 3))
   input_tensor= keras.Input(shape=(320, 320, 1))
2  x = keras.layers.Conv2D(3, (3, 3), padding='same')(input_tensor)
3  x = mobilenet(x)
4  y = keras.layers.GlobalAveragePooling2D()(x)
5  y = keras.layers.Dense(14, activation='sigmoid')(y)
6  model = keras.Model(input_tensor, y)
7  model.compile(tf.keras.optimizers.Adam(learning_rate=learning_rate),
   loss=keras.losses.BinaryCrossentropy(),
   metrics=[keras.metrics.BinaryAccuracy()])

8  history = model.fit(train_generator, verbose=1, epochs=epochs, batch_size= 512,
   shuffle=True)
```

Algorithm 5: DenseNet with Transfer Learning

Input: epochs, learning rate, and train_generator.
Output: loss, metrics, and history.

```
1  densenet = keras.applications.DenseNet(include_top=False,
   weights='imagenet',input_shape=(320, 320, 3))
   input_tensor= keras.Input(shape=(320, 320, 1))
```

```
2    x = keras.layers.Conv2D(3, (3, 3), padding='same')(input_tensor)
3    x = mobilenet(x)
4    y = keras.layers.GlobalAveragePooling2D()(x)
5    y = keras.layers.Dense(14, activation='sigmoid')(y)
6    model = keras.Model(input_tensor, y)
7    model.compile(tf.keras.optimizers.Adam(learning_rate=learning_rate),
     loss=keras.losses.BinaryCrossentropy(),
     metrics=[keras.metrics.BinaryAccuracy()])

8    history = model.fit(train_generator, verbose=1, epochs=epochs, batch_size= 512,
     shuffle=True)
```

Algorithm 6: ResNet with Transfer Learning

Input: epochs, learning rate, and train_generator.
Output: loss, metrics, and history.

```
1    resnet = keras.applications.ResNet(include_top=False,
     weights='imagenet',input_shape=(320, 320, 3))
     input_tensor= keras.Input(shape=(320, 320, 1))
2    x = keras.layers.Conv2D(3, (3, 3), padding='same')(input_tensor)
3    x = mobilenet(x)
4    y = keras.layers.GlobalAveragePooling2D()(x)
5    y = keras.layers.Dense(14, activation='sigmoid')(y)
6    model = keras.Model(input_tensor, y)
7    model.compile(tf.keras.optimizers.Adam(learning_rate=learning_rate),
     loss=keras.losses.BinaryCrossentropy(),
     metrics=[keras.metrics.BinaryAccuracy()])

8    history = model.fit(train_generator, verbose=1, epochs=epochs, batch_size= 512,
     shuffle=True)
```

In terms of running the code for the training process, access to a remote server on the university’s supercomputer was required. This is because training thousands of images requires high GPU power, in order for the code to run fast and for training time to be shortened.

The metrics, accuracy and loss, were used to assess how the hyperparameters should be finetuned. The behavior of the loss function was observed, along with the final training accuracy, in order to determine optimal values for the learning rate and the batch size. As such, the training process was conducted many times. In each time, the hyperparameters were adjusted, until the final training accuracy was high and the loss was low.

2.2.2 Testing

The testing stage was conducted after the training has been completed, for each model. Once we have a fully trained model, we can begin to test it on the dataset that we have set aside previously for testing. Since the trained model has not seen these images before, this will test its ability to accurately predict the label associated with each one. The testing process revealed the overall loss computed by the loss function. In addition to that, it will reveal the probabilities associated with each class for each given image. This will allow us to assess our algorithm and determine how we can fine tune it to yield lower losses and more confident probabilities.

During the testing process, two important functions were used: "model.evaluate" and "model.predict." The former was used to determine the overall accuracy and loss of a model, given the testing dataset. On the other hand, the latter was used to output the classification for each given image of a chest x-ray. Both of these functions were useful in representing how well the model was able to generalize on new data.

2.3 Quantitative Assessment of Algorithms

The performance of Che-X-Ray can be determined by observing the performance criteria of machine learning algorithms. We will begin by defining the loss function, which is an important function that determines how effectively a machine learning system performs. Then, we will show how these loss functions can be observed in Python during the testing of the algorithm.

A loss function is a mechanism for determining how effectively a machine learning system predicts the featured data set. In other words, loss functions are a measure of how well your model predicts the predicted outcome.. The loss function is closely tied to the model prediction that a user created. The model will produce decent results if the loss function value is low. Machine learning does not have one specific one-size-fit loss function. Each of the machine learning algorithms we will be employing have a certain loss function that they use to determine their performance.

Furthermore, to quantitatively assess and compare the efficacy of the proposed algorithms, the binary accuracy will be tracked for each epoch in the training process. Finally, at the end of the training, the overall binary accuracy will be recorded. Furthermore, in the testing stage, the binary accuracy of the test images will also be taken and recorded. As such, these two metrics, binary training accuracy and binary testing accuracy, will be used to compare the accuracy of each algorithm.

2.4 Final Algorithm

After conducting the quantitative assessment for each of the algorithms, the most accurate algorithm was chosen to be the final model. During each training process, the model parameters, the weights and biases, were saved for each of the algorithms. This means that after picking the model that will be used, the parameters could be easily called, so the training process does not have to be completed each time.

Then, a short piece of code was written, which was comprised of three parts: loading the most accurate model and its parameters, prompting the user to input an image, and finally giving an output that lists the pathologies present in the chest x-ray image. This short code will be embedded into the GUI, so that the radiologist can access the algorithm.

2.5 Graphical User Interface

The next step was the creation of a graphical user interface that the radiologist can use. The GUI is essentially a website created using WordPress. The website includes a homepage, which has hyperlinks to basic features such as: an about us page, a contact page, and a FAQ page.

Once the homepage was created, Javascript was used to apply a web interface on the algorithm written on Python. Javascript has Tensorflow packages, which essentially means we can deploy our trained model on the browser. After writing the Javascript code, it was run on the browser through an HTML piece of code. In that way, a link is generated, so that any user would be able to access it on the browser.

This was hyperlinked in the homepage of the website, so so that a radiologist would be able to click on the link, upload the x-ray image, and get a classification based on the model's predictions. As such, the complexity of the algorithm will be hidden, meaning the radiologist will only be able to see the output once it has been predicted by the algorithm.

3. RESULTS

3.1 AI Algorithm

3.1.1 Training

The training results are comprised of two important metrics: binary training accuracy and the loss (binary cross entropy). These metrics were observed for the two types of training techniques that were implemented: training from scratch and training using transfer learning (employing a pre-trained model).

In terms of the results for the training from scratch, all the models, DenseNet, MobileNet, and ResNet had very low training binary accuracies, reaching a max of only 83%, no matter how much the models' parameters were fine-tuned.

On the other hand, the transfer learning technique yielded better results. Each pre-trained algorithm and their binary accuracies and losses can be summarized in Table 1.

Table 1: Training Results

Pretrained CNN Models	Training Accuracy	Training Loss
MobileNet	89.93%	1.34
DenseNet	90.21%	0.79
ResNet	88.92%	0.98

3.1.2 Testing

The testing was supposed to be conducted on a private dataset that we obtain from local hospitals. While approval was obtained from the hospital to get the chest x-ray images, approval from the IRB was difficult to obtain. It is important to note that this private dataset was not collected for this research, but they existed at a local hospital for other research. However, in order for us to receive the dataset, an IRB from our end had to be filed and submitted. Unfortunately, obtaining a TAMU IRB required a longer time period than this project allowed. Thus, we were not able to obtain nor use the private dataset for this project. Thankfully, we had a diverse and large public dataset that was sufficient to fulfill the goals set for this project. Thus, the testing process was conducted on 20% of the public dataset that was set aside for testing.

On another note, the testing results are comprised of two important metrics: binary testing accuracy and the loss (binary cross entropy). As with the training, the results from scratch technique yielded very low testing accuracies for all of the models. The testing accuracies did not surpass 45%, no matter how much the models' parameters were fine-tuned.

On the other hand, the transfer learning technique yielded better results. Each pre-trained algorithm and their binary accuracies and losses can be summarized in Table 2.

Table 2: Testing Results

Pretrained CNN Models	Training Accuracy	Training Loss
MobileNet	84.52%	1.72
DenseNet	87.54%	1.03
ResNet	81.39%	1.49

3.2 Graphical User Interface

A webpage was created using a system software called WordPress, in order to allow radiologists to upload their data or images. The homepage of the website can be seen in Figure 4 below.

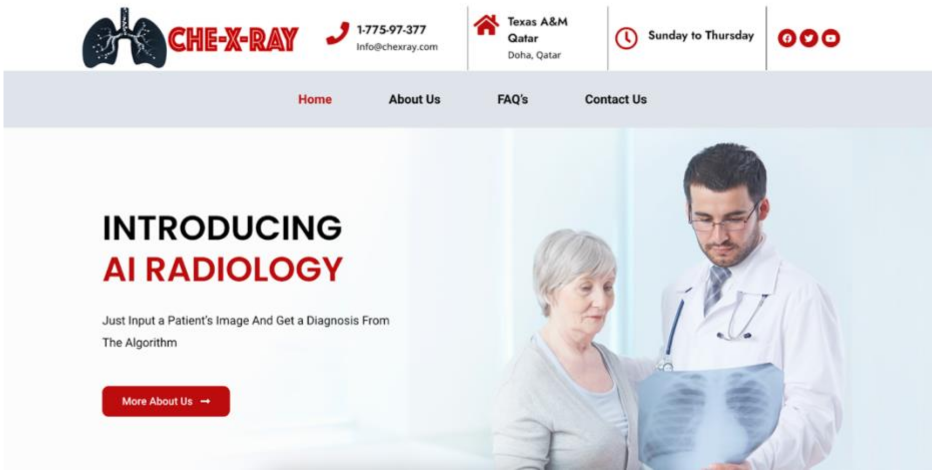

Figure 4: GUI homepage.

Clearly, the homepage includes basic features, such as: about us page, FAQ page, and a contact page. Furthermore, there is also a hyperlink to the algorithm, where the radiologist can use the AI to help them diagnose their patient.

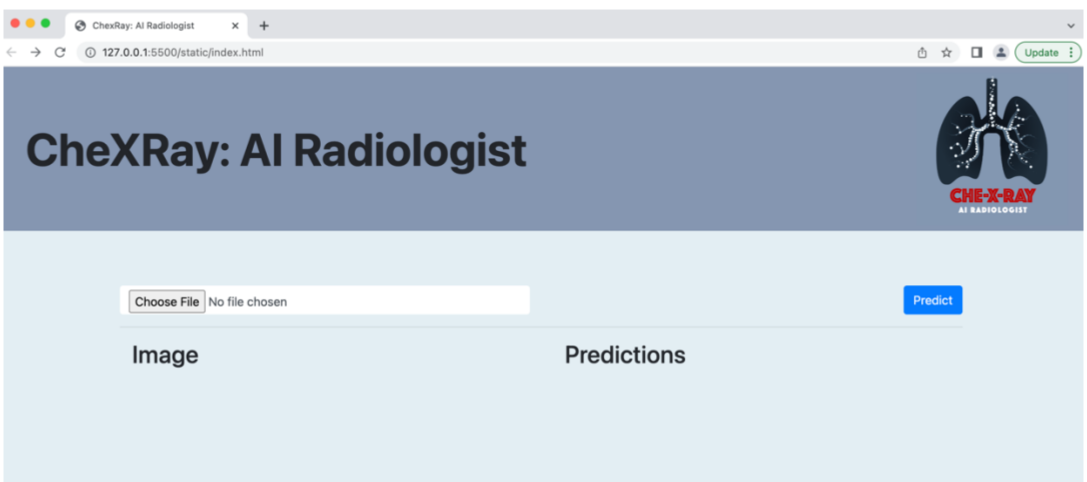

Figure 5: GUI image upload page.

When the hyperlink is clicked, a webpage is launched that looks like Figure 5. The user, the radiologist, will then be prompted to choose a file from their local directory. The file in this case should be the x-ray image that the radiologist needs help with diagnosing. Then, after picking the file, the image will appear on the left side of the screen. The user can then click the predict button on the right side of the screen to get the top diseases associated with the image, as well as their corresponding probabilities. The algorithm only outputs the diseases that have a probability greater than 0.5, because that may indicate a positive diagnosis for a particular disease. Thus, after clicking the predict button, the webpage should look like Figure 6.

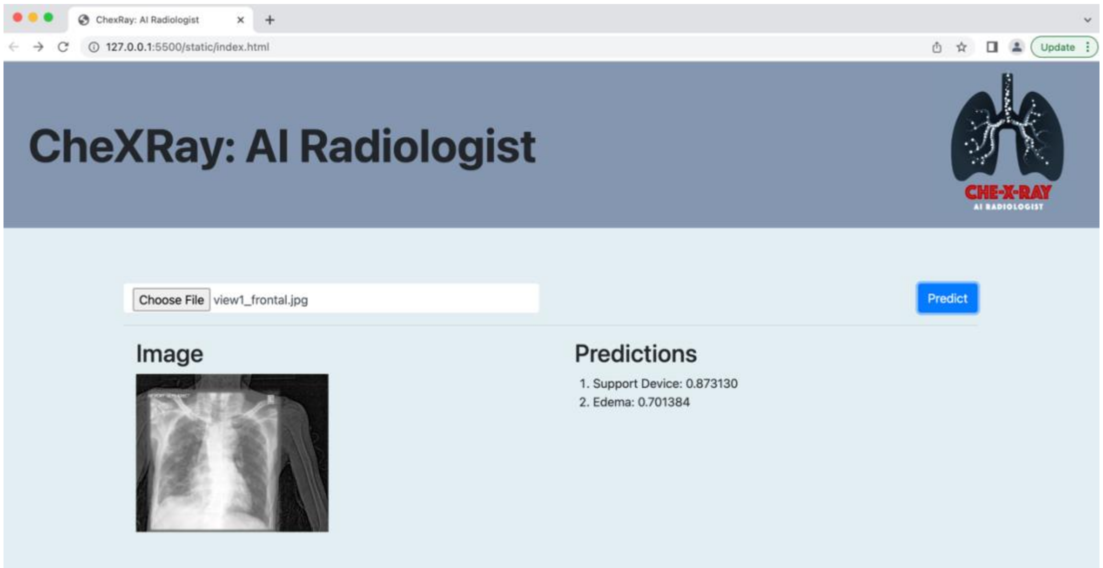

Figure 6: GUI image and results.

The entire process can be summarized as follows: the radiologists uploads a chest x-rays image that they would like to analyze and get a classification for. Then, they click on the predict button on the upper right corner to get an output, which is essentially the top diseases associated with the x-ray image, as well as their respective probabilities.

4. CONCLUSION

4.1 Discussion

Initially, training from scratch was implemented. With this technique, training accuracy had a maximum of 83% no matter what model was being implemented. The reason the accuracy topped at this value, no matter how much the hyperparameters were tuned, was because of the lack of resources in training. In order to be able to train huge models from scratch, high training time and more GPU power are needed to train. However, those resources were not available to use for this project, and for that reason transfer learning was the better solution.

On the other hand, training using transfer learning yielded better results. Tables 1 and 2 display the training and testing binary accuracies for the three networks used in the project. The highest training and testing accuracy was from using DenseNet with 90.21% and 87.54% respectively. The accuracies can be improved by adding more images to the dataset and fine tuning the parameters. Overall, there is improvements when resorting to transfer learning rather than training from scratch. This helped in obtaining more accurate results and having a faster overall training time.

In terms of the private dataset, images of chest x-rays were supposed to be obtained from the local hospitals in Qatar. However, an IRB to receive and use these images was not received on time to be able to implement this part of the project. Fortunately, 20% of the public dataset was reserved for testing, in case this occurred. This implies that the models that were trained and tested are not localized to Qatar, but more general. While the initial plan was to localize the model, so that it is able to make generalizations in this region, it is still fortunate that the model is able to make predictions on a diverse set of data.

4.2 Analysis

First and foremost, AI is becoming increasingly prevalent in almost every field. It is beginning to have adaptations in healthcare, education, and medicine. When it comes to the medical field, the AI algorithm that is proposed and implemented by this project is intended to be used as a tool rather than a replacement for a human doctor. This idea has recently been a point of ethical debate, which is why it is important to address. For example, the same way a doctor could consult another doctor for a second opinion on an x-ray, the doctor could use this tool to help it form its diagnosis. What is particularly useful about this project is that for every pathology it predicts a patient would have, there is a probabilistic score indicating how confident the algorithm is that a certain pathology is evident in the picture. This feature kind of objectifies an aspect of radiology that we would have deemed somewhat subjective in the past. Usually, when radiologists give their opinion, there is no measure of how confident they are in their diagnosis. Therefore, this tool kind of allows to radiologist to guage how confident they should be in a particular diagnosis.

4.3 Future Plans

For future plans, our primary goal is to improve the accuracy of our data analysis. One approach to do this is to fine-tune the parameters the models. Furthermore, we must first acquire clearance from an Institutional Review Board (IRB) before testing our algorithms on private data. This will help to conduct research that is both ethical and compliant. Lastly, we want to keep improving our models and approaches.

REFERENCES

[1] "Chexpert: A large dataset of chest x-rays and competition for automated chest X-ray interpretation.," *CheXpert: A Large Chest Radiograph Dataset with Uncertainty Labels and Expert Comparison*. [Online]. Available: https://stanfordmlgroup.github.io/competitions/chexpert/. [Accessed: 20-Aug-2022].

[2] A. Shenoy, "Pseudo-labeling to deal with small datasets - what, why & how?," *Medium*, 03-Dec-2019. [Online]. Available: https://towardsdatascience.com/pseudo-labeling-to-deal-with-small-datasets-what-why-how-fd6f903213af. [Accessed: 21-Sep-2022].

[3] R. Chhugani, "Most used loss functions to optimize machine learning algorithms," *Analytics India Magazine*, 07-Oct-2021. [Online]. Available: https://analyticsindiamag.com/most-used-loss-functions-to-optimize-machine-learning-algorithms/. [Accessed: 21-Sep-2022].

[4] "Understanding Categorical Cross-Entropy Loss, Binary Cross-Entropy Loss, Softmax Loss, Logistic Loss, Focal Loss and all those confusing names," *GitHub*, May-2018. [Online]. Available: https://gombru.github.io/2018/05/23/cross_entropy_loss/. [Accessed: 21-Sep-2022].

[5] J. Frost, "Mean squared error (MSE)," *Statistics By Jim*, 14-Nov-2021. [Online]. Available: https://statisticsbyjim.com/regression/mean-squared-error-mse/. [Accessed: 21-Sep-2022].

[6] J. Brownlee, "What is a confusion matrix in machine learning," *Machine Learning Mastery*, 14-Aug-2020. [Online]. Available: https://machinelearningmastery.com/confusion-matrix-machine-learning/. [Accessed: 21-Sep-2022].