2021-12-01

A Integrated Approach Of Deep Learning And Augmented Reality For Pneumonia Detection In Chest X-Ray Images

Jeevarathinam Senthilkumar
University of Texas at El Paso

A INTEGRATED APPROACH OF DEEP LEARNING AND AUGMENTED REALITY FOR PNEUMONIA DETECTION IN CHEST X-RAY IMAGES

JEEVARATHINAM SENTHILKUMAR

Master’s Program in System Engineering

APPROVED:

Tzu-Liang (Bill) Tseng, Ph.D., CMfgE, Chair

Md Fashiar Rahman, Ph.D.

Eric D. Smith, Ph.D.

Yirong Lin, Ph.D.

Stephen L. Crites, Jr., Ph.D.
Dean of the Graduate School

A INTEGRATED APPROACH OF DEEP LEARNING AND AUGMENTED REALITY FOR PNEUMONIA DETECTION IN CHEST X-RAY IMAGES

by

JEEVARATHINAM SENTHILKUMAR

THESIS

Presented to the Faculty of the Graduate School of
The University of Texas at El Paso
in Partial Fulfillment
of the Requirements
for the Degree of

MASTER OF SCIENCE
Department of Industrial, Manufacturing and Systems Engineering
THE UNIVERSITY OF TEXAS AT EL PASO
December 2021

ABSTRACT

Pneumonia is a viral or fungal illness that spreads to the lungs of the human body, causing fluid to accumulate in the lungs' air sacs. Millions of people are affected by this disease each year. One of the most common radiological diagnostics for diagnosing and screening this kind of sickness is a chest X-ray. The most commonly available radiological test for diagnosing and screening this kind of illness is a chest X-ray. An inaccurate diagnosis, especially over-diagnosis and under-diagnosis, is a common issue in the medical sector. As another issue, human-assisted diagnosis has limitations like the availability of an expert, cost, etc. To address these issues, researchers focused on deep learning approaches for improved diagnostic outcomes. An application with an automatic system to detect pneumonia is developed in these aids in overcoming the diagnosing errors and treating the patient. As discussed above, the authors developed a two-step methodology in this research. In the first step, various models are utilized as the neural network model to be trained with different software for pneumonia detection for chest X-ray images. These trained models are converted to a frozen graph and injected into the Unity software tool by utilizing a C# script that creates a bounding box to overlay the detected region. These processes are mitigated by building and deploying the application from unity to the head-mounted device known as Microsoft HoloLens gen 1. This study also considers the research gap and proposes integrating Augmented Reality (AR) technology and Deep Learning (DL) to handle this issue.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

Pneumonia is a bacterial infection or fungi on one or both lungs, an acute respiratory disease. It is one of the major life-threatening diseases of people of all ages. A recent study from the World Health Organization (WHO) states that the mortality rate of 808,000 children is under the age of 5 until 2017. Even people above the age of 65 are also highly at risk of death. Every year the death rate due to this disease increases by 18%. One of the main reasons for these diseases is overweight and the use of tobacco. Based on the WHO report, at least 8% of heart disease, diabetes, and 40% of cancer can be reduced to the avoidance of tobacco.

This disease can be diagnosed by one of the best-known medical imaging, known as Chest X-Ray. Chest X-Ray imaging is also used to diagnose different heart problems, collapsed lung, broken ribs, emphysema, and lung cancer. In specific, Pneumonia can be detected based on the area of increased opacity on a chest x-ray. Yet, the imaging reviews on Chest X-rays are complicated because several factors, such as positioning of the patient and depth of inspiration, can alter the appearance of chest X-Ray.[1] Even the expert radiologist still makes mistakes sometimes because observing X-Ray images is still considered subjective based on an institution.

In recent years, the growth of computer advancement diagnostic and image acquisition devices and the medical image sets is also increasing exponentially for diagnoses. Both clinical science and medical treatments also benefit from the advanced digital processing and content storing system. A large amount of data is available by different hospitals and diagnostic centers. To analyze these data, several algorithms have been proposed. The Convolutional Neural Network (CNN) has been a significant advancement for medical image classification.[2] One of the most popular networks learns by extracting discriminative features necessary for classifying an image.

In this study, the MobileNets, a Convolutional Neural Network, is trained on a dataset with two different classes: Pneumonia and normal. Later, the trained model is imposed into a Unity framework to create an Augmented Reality mobile application as a TensorFlow Lite Model.[3] This application will aid radiologists and medical experts in efficient diagnosis.

1.1 What is Pneumonia?

The infectious or non-infectious diseases caused in the lungs may lead to pneumonia. Pneumonia is considered the sixth leading cause of death in The United States of America. A recent report on the centers for disease control and prevention states that out of 1.1 million people, more the 50,000 were dead due to pneumonia disease. Despite having the total number of deaths due to lower respiratory tract infections available, there is no systemic study conducted on the incidence of pneumonia. Pneumonia can be classified into different types; based on Anatomical Classification – Bronchopneumonia, Lobar Pneumonia, Segmental Pneumonia, and Sub Segmental Pneumonia. Microbiologist or radiologist – bacterial, viral, fungal, chemical pneumonia, Parasitic pneumonia, and physical pneumonia or ionizing pneumonia. Caustic organism-based classification of pneumonia is more widely used when compared to the anatomical-based sort of pneumonia.[4] For diagnosis, there are some conditions when a lesion in the lung is diagnosed as pneumonia, but it fails to respond to treatment, or instead, it responds very late; in this condition, this is labeled as non-resolving pneumonia. Different physicians around the globe diagnose this type of pneumonia by their own opinion, for example, based on radiological infiltration not resolving, the due time course of ten days with antibiotic therapy.

Physicians diagnose this type of disease by asking the patients about their living circumstances, occupation, history of travel, contact history with patients, and exposure to animal history, which may get clues about the microbial etiology of the infection.[5] The clinical and radiological methods may help in concluding the diagnosis of pneumonia. And the laboratory infection will help to complete the physicians about the etiology of the disease.

1.2 Different Diagnosing Methods

There are two standard types of diagnosing methods is followed in the medical industry—clinical and radiological diagnosis. Infectious diseases and noninfectious diseases such as chronic bronchitis - acute exacerbations, acute bronchitis, radiation pneumonitis, pulmonary embolism, and heart failure are community-acquired pneumonia (CAP) diagnoses. Meticulous history collecting of a patient is more important. For example, they are worsening pulmonary edema suggested by known cardiac disease, Secondaries to irradiation therapy presented by underlying carcinoma. Epidemiologic clues, like a patient's recent travel history to available endemic areas, for example, travel to southeast Asia, may alert the physician to reach the specific possibilities. Moreover, the specificity and sensitivity of the physical examination findings are significantly less as an averaging sensitivity of 58% and specificity of 67%.

Since there is very little sensitivity and specificity in physical findings is necessary to differentiate CAP from the other conditions by using a chest X-ray.[6] A sample figure of chest X-ray is given below (See figure 1.2.1). Radiological findings such as TB, cavitation in the upper lobe, and Staphylococcus aureus infection indicated by pneumatoceles can sometimes imply the etiological diagnosis.[4] In outpatients, treatment for CAP is done with the help of clinical and radiologic assessments because laboratory results are often unavailable to start initial treatment. In some cases, the availability of rapid diagnosis and treatment is critical. For influenza virus infection, prompt diagnosis is vital to begin anti-influenza treatment and secondary prevention of disease. Usually, CT is unnecessary, but sometimes in a patient with a foreign body or tumor-causing post obstructive pneumonia, it is more beneficial to diagnose.

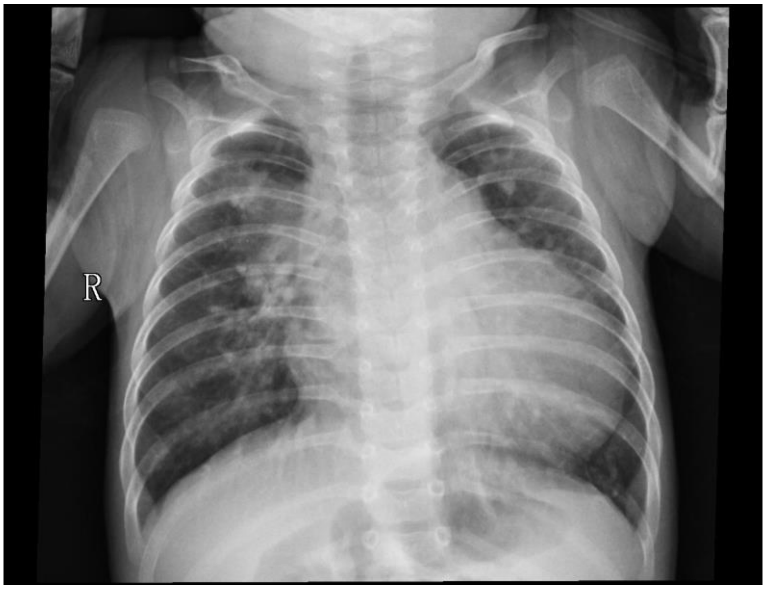


Figure 1.2.1: Chest X-ray image of an infected patient

The profound finding of radiological diagnosis and its technical aspects are explained below. Turn, motive, and entry are three parameters that are frequently assessed to confirm the acceptable quality of any chest X-ray. These days, infiltration via modern computerized tactics is less significant. An adequate chest X-ray should have no course. This is validated by confirming that the clavicle's medial margins are equidistant from the vertebral bodies' spinous processes. It's also crucial to make sure the patient is making a solid attempt to breathe.[7] On the frontal chest X-ray, eleven ribs can be seen posteriorly near the midclavicular line. It's critical to consider why these two technical elements are so significant at this point. Effectively, the answer is that failure to ensure the correct standard can mimic life-threatening pathology in each case. Pneumonia is known as the increased darken region. As a result, one side darkens more than the other. The reviewer may misunderstand this variation in the shaded area as pathology, which would be a diagnostic error.[8] In-depth, the darken region represents a pneumothorax or pulmonary embolism, and the lighter part may be mistaken as a pleural fluid collection or air space consolidation. This is particularly likely on a supine radiograph.

The radiologist reviews the chest X-ray by confirming the patient details are correct, and the chest X-ray is technically satisfactory to check in a systematic order.[9] The chest X-ray is labeled using an international standard known as posteroanterior PA, which has no annotations, and anteroposterior AP is marked for reviewer help.[10] Nonetheless, the radiologist is advised not to start a report on PA or AP because there might be a slew of problems if the diagnosis isn't reported correctly.[6] For instance, failure to recognize a second breast shadow suggests an underlining diagnosis of mastectomy and, therefore, should prompt a systematic review for evidence of breast cancer and its sequelae. These complications faced by expertise may lead to slow down the treatment process for the patients.

1.3 Usage of Modern Technology for Diagnosis

With the growth of computers advancement diagnostic and image acquisition devices in recent times, medical image sets are also increasing exponentially. Clinical science and medical treatments also benefit from the advanced digital processing system and produce large amounts of data. Therefore, deep analysis of the diseases is also a challenging issue. Several techniques have been suggested for analyzing these large data sets. Deep learning-based classification is one of the efficient techniques for analyzing medical images.[11] Deep learning techniques can achieve near to optimal solutions in practical aspects due to their end-to-end learning capability.

1.4 Which Modern Technology is considered Better?

As per the two-step methodology, the first step is undergone by training a neural network model to detect pneumonia from a chest X-ray image. Based on the survey, the image from the dataset is utilized for training using machine learning and deep learning algorithms. The deep learning algorithms with better results are considered based on the impact and working nature of the model in the unity framework. The model is converted into a frozen graph where the unity script can understand and interact to act. The detailed structure of the training and conversation of the model is described in chapter 5. A unity scripting is developed to make the interaction between the model and the application. The solution is considered better based on the performance of the multithreaded process, the process of running multiple programs of getting and set to show the result simultaneously with not much delay. The detailed structure of the training and conversation of the model is described in chapter 5.

CHAPTER 2: LITERATURE SURVEY

Rahamn et. al This paper proposes a new method to focus on the disease-specific attention region. The Pneumonia portion of a picture is erased and designated as a non-pneumonia sample using this procedure. And to decrease background interference, transfer learning is employed to segment the interest area of the lungs. This dataset is trained using SE-ResNet Convolutional Neural Network.[12]

This paper proposed to generate the medical images by utilizing Convolutional Data Augmentation and Generative Adversarial Networks (GANs).[13] This Generative Adversarial Networks produced unprecedented Chest X-Ray images of patients trained on the DCNN model to classify the images. The utilizing of synthetic data produced by the GAN model increased the accuracy.

This paper compares the four models: primary convolutional neural network, VGG16, VGG19, InceptionV3 to detect pneumonia from chest X-Ray images. These models are constructed using CNN and Transfer Learning methodology and produce accuracy of over 97 percent from all models.[3]

The Deep Convolutional Neural Network is designed for multiclass classification to classify fifteen diseases from chest X-Ray images. This model consists of convolutional layers, ReLU Activations, Pooling layer, and fully connected layer.[2] This last fully connected layer consists of fifteen output units, where each output unit has a probability to predict one of fifteen diseases. The average accuracy achieved from the classification of different diseases is 89.77%.
This paper proposes the two well-known convolutional Neural Network models Xception, and VGG16, to diagnose pneumonia from chest X-Ray images. [14]For training Transfer learning, fine-tuning methods are used, with 0.87% and 0.82% for VGG16 and Xception models.

CHAPTER 3: BACKGROUND

This Chapter discusses the overview about the fundamentals topics of platform and devices used for this thesis, such as Neural networks, machine learning, and the head-mounted device (Microsoft HoloLens Gen 1)

3.1 NEURAL NETWORK

Deep Learning is a supervised learning technique that employs NNs, commonly known as Artificial Neural Networks (ANNs). NNs are inspired by the human brain and use the concept of neurons. Early computational models for artificial neurons were proposed in 1943 by Warren S. McCulloch and Walter Pitts. An early NN called the perceptron was developed by Frank Rosenblatt in 1957. In 1969 Arthur Bryson and Yuchi described backpropagation as an optimization method used to train NNs. However, the advent of distributed computing and powerful Graphics Processing Units (GPUs), which enabled the usage of larger NNs, caused the popularity of NNs and Deep Learning in recent years.[15] Neurons have one or multiple inputs as well as an output value which is called activation. Each piece of information has a weight assigned to it. A bias value is assigned to each neuron. These and the input values are added together in a weighted total. Equation 3.1.1 is an example of such a weighted sum. The input value x of each input is weighted using the corresponding weight w. To this weighted sum, the bias b of the neuron is added.

Equation 3.1.1 $$w0.x0 + w1.x1 + w2.x2 + \cdots + wn.xn + b0$$

The weighted sum is put into an activation function to normalize the value. The sigmoid function Equation 3.1.2 used to be the most common one.[16] However, other functions such as Rectified Linear Unit (ReLU) Equation 3.1.3 or tanh are also possible. ReLU is nowadays the most common one.

Equation 3.1.2 $\sigma(z) = 1/(1 + e^{-z})$

Equation 3.1.3 $\sigma(z) = max(0, z)$

Multiple neurons form a layer. The outputs of one layer's neurons are coupled to the inputs of the next layer's neurons. A NN always has an input layer that receives external data and an output layer that produces the eventual result. Between the input and output, layers are zero or more so-called hidden layers.

Weights and activation values are the values that are learned during the training of the NN. For all training inputs x, the training aim is to identify values where the NN output approximates y1xo (the intended outcome of a training example). A cost function is used to assess how effectively the neural network performs its job. A cost function like Equation 3.1.2 exists.

b is a collection of the biases, w of weights, n is the number of training pairs, a is the vector of all output values for an input x, and the sum is over all training inputs x. The objective is to develop weights and biases that will allow the cost function to be minimized. This may be accomplished via the use of a backpropagation method that employs gradient descent. The algorithm iterates backward layer-by-layer from the last layer while computing the loss function's gradient, respecting each layer's weights. The primary networks where the outputs from one layer are used as input for the subsequent layer are called feedforward neural networks. NNs with multiple hidden layers are often called Deep Neural Networks (DNNs).[17] Information only flows forwards and never backward since there are no loops. However, some NNs allow having feedback loops. Such NNs are called Recurrent Neural Network (RNN).

3.1.1 Convolutional Neural Network

Another type of NNs, which are crucial for image classification and pattern recognition tasks, the filter is Convolutional Neural Networks (CNNs).[1] These NNs feature two additional types of layers: convolutional layers and pooling layers. The former calculates the output of neurons connected to the input regions using convolution: the filter kernel glides step-by-step over the input. The output value is calculated using the scalar product of the kernel and the underlying section of the information. An activation function, usually ReLU (Equation 2.3), is applied to calculate the neuron’s final activation value. The values of the filters themselves are determined during training.[18] A convolutional layer can have multiple filters which extract different features. Each filter results in a separate feature map. After the convolutional layers follow pooling layers which perform downsampling and reduce the number of parameters, pooling layers are sometimes also called subsampling layers. The most common method is max pooling which only keeps the maximum value of each 2x2 square. An alternative approach is average pooling, where the average value of each 2x2 square is used.

Both techniques reduce the data by 75%, which increases the performance and helps against overfitting. Lastly, CNNs also contain ordinary, fully connected layers like traditional NNs. For classification tasks, the number of neurons usually corresponds to the number of classes. CNN’s can have multiple convolutional and pooling layers and multiple fully connected layers at the end. Pixel values of images are usually represented as an 8-bit integer, whereby the values range from 0-255.[12] Many CNNs expect a range of 0-1. Therefore, the values must be normalized accordingly. The weight values are usually smaller than one, and significant integer inputs can slow down or rattle the training process.

3.2 Machine Learning

Another type of NNs, which are crucial for image classification and pattern recognition tasks, the filter is Convolutional Neural Networks (CNNs). These NNs feature two additional types of layers: convolutional layers and pooling layers. The former calculates the output of neurons connected to the input regions using convolution: the filter kernel glides step-by-step over the input. The output value is calculated using the scalar product of the kernel and the underlying section of the information. An activation function, usually ReLU (Equation 2.3), is applied to calculate the neuron's final activation value. The values of the filters themselves are determined during training.[10] A convolutional layer can have multiple filters which extract different features. Each filter results in a separate feature map. After the convolutional layers follow pooling layers which perform downsampling and reduce the number of parameters, pooling layers are sometimes also called subsampling layers. The most common method is max pooling which only keeps the maximum value of each 2x2 square. An alternative approach is average pooling, where the average value of each 2x2 square is used.

Both techniques reduce the data by 75%, which increases the performance and helps against overfitting. Lastly, CNNs also contain ordinary, fully connected layers like traditional NNs.[19] For classification tasks, the number of neurons usually corresponds to the number of classes. CNN's can have multiple convolutional and pooling layers and multiple fully connected layers at the end.[11] Pixel values of images are usually represented as an 8-bit integer, whereby the values range from 0-255. Many CNNs expect a range of 0-1. Therefore, the values must be normalized accordingly. The weight values are usually smaller than one, and significant integer inputs can slow down or rattle the training process.

3.3 Mixed Reality

In Virtual Reality (VR), users will get immersed in an entirely synthetic world. VR and the natural world are not antitheses; instead, they are on opposite ends of a continuum. This concept, which was introduced by Milgram et al. in 1994, is called the Reality-Virtuality continuum. The area between VR and the natural world is called MR.[20] It contains AR and augmented virtuality, which means that natural world objects are integrated into virtual worlds. AR is about enriching the real world with virtual objects. This includes overlaying additional information over the real world using a smartphone or AR Head-Mounted Displays (HMDs) such as Google Glass.[21] However, overlaying information on video streams showing the real world on a computer screen is also AR. In practice, the differences between AR and MR are not always apparent since both terms are sometimes used as synonyms. However, MR includes more than just AR according to the Reality-Virtuality continuum.[20] A sample diagram represents the description of MR (See Figure 3.3.1). Microsoft uses the term MR for experiences where virtual 3D holograms are inserted in the real world and coexist with physical objects. An example would be virtual models of furniture placed into actual rooms to see how they fit.[22] This requires sensors or cameras which detect the location of the MR device and nearby objects.

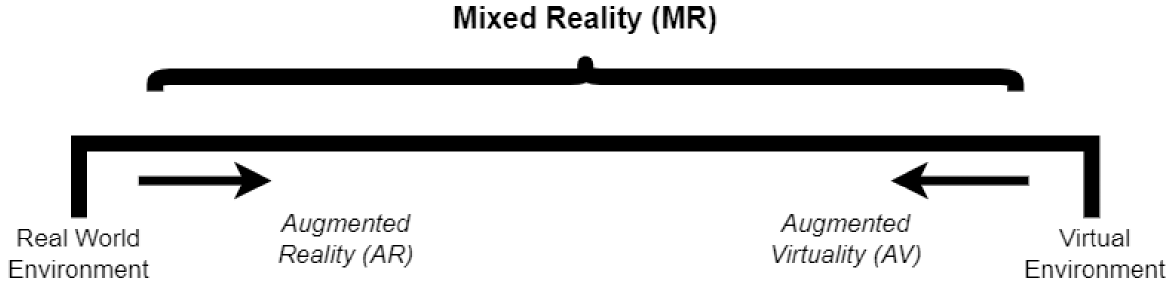


Figure 3.3.1: Mixed Reality technology

3.4 MICROSOFT HOLOLENS

Microsoft's HoloLens is a mixed reality headset. See-through holographic lenses are used to achieve MR. The gadget may be controlled with gestures, speech, and eye-tracking. It is self-contained due to the presence of a computer.[23] Universal Windows Platform is the operating system. The HoloLens comes in two generations.

3.4.1 Microsoft HoloLens Gen 1

The first generation of the HoloLens (shown in figure 3.4.1.1) was released in 2016. According to Microsoft, it is the first untethered AR headset in the world. The internal computer is based on an Intel 32-bit architecture. Furthermore, it incorporates a Holographic Processing Unit (HPU), a custom-made coprocessor for processing sensor values and holograms. The HoloLens (1st gen) is now in Long Term Servicing state. Therefore, it will not receive further updates other than bug and security fixes.[24]

Table 3.4.1.1: Microsoft HoloLens Gen 1 Specification and details

Specification	**Details**
CPU	Intel 32 bit
Memory	2 GB RAM, 1 GB Holographic processing unit
Storage	64 GB
Display resolution	1280x720
Field of View	34^o
Microphone	Four channel arrays
Camera	2.4 MP, HD video
Hand Tracking	Both hands
Weight	579 g

The October 2018 update with build number 1809 is the latest available version for the HoloLens (1st gen). The most severe limitation of the old Windows version is that the HoloLens (1st gen) can only Universal Windows Platform (UWP) applications and no ordinary Win32 applications.[24] Furthermore, up-to-date browsers such as the Chrome-based Edge browser are not available.

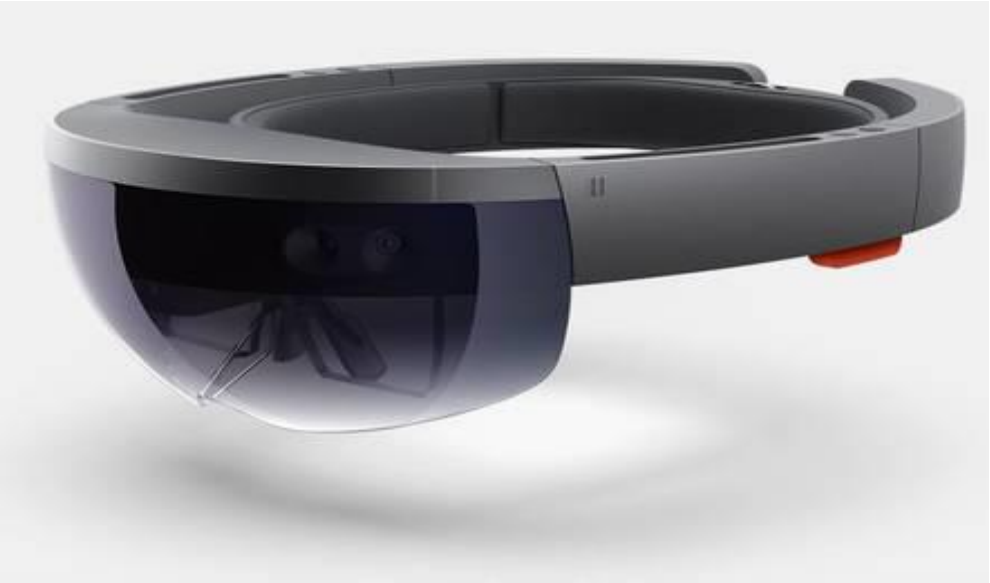


Figure 3.4.1.1: Microsoft HoloLens Gen 1

CHAPTER 4: SOFTWARE

This section covers the different software tools, frameworks, datasets, and models used in this thesis, including multiple machine learning frameworks and graphics engines.

4.1 TENSORFLOW

TensorFlow is a framework for dataflow programming developed by Google. The primary use is machine learning, with a focus on neural networks. It was published as an open-source program in 2015. TensorFlow replaced Google's Disbelief project. Directed graphs consisting of nodes describe TensorFlow computations. These directed graphs represent dataflow computations. Each node instantiates an operation and has zero or more inputs and zero or more outputs. Tensors are multidimensional arrays that are passed on edges from outcomes to inputs. Operations represent abstract computations, e.g., add or matrix multiply. Operations can have attributes that have to be provided at construction time. Kernels are implementations of operations for specific types of devices such as CPUs or GPUs. Client programs interact with TensorFlow by creating a session.[25] The session interface provides a run method that invokes the computations and a comprehensive approach to augment other nodes and edges to the current graph. A critical feature of TensorFlow to point out is the automatic gradient computation which is required by many ML training algorithms. For instance, the minimization of the cost function during the training of NNs makes use of gradient computation.

TensorFlow's low-level Application Programming Interfaces (APIs) are used to design computational graphs. The API is available in several programming languages, including Python and C++. For creating NNs, TensorFlow includes the high-level Keras API. In short, Keras provides an abstraction layer for defining NNs by its layer architecture. TensorFlow's computational graphs are still used under the hood. Section 4.2 describes Keras in more detail.

TensorFlow is available for several platforms, including x86_64 systems running Linux or Windows, Android, Raspberry Pi, and more. GPU acceleration is also supported on CUDA (mainly GPUs from Nvidia) enabled GPUs. Furthermore, community builds also support AMD ROCm capable GPUs.[25]

In 2019 TensorFlow 2.0 was released. The new version integrates Keras more tightly. Furthermore, the low-level API offers more possibilities to access the internal functions of TensorFlow. The Saved Model file format is more standardized and will replace the different forms of TensorFlow Lite and TensorFlow.js in the future. There are also several improvements in terms of training. TensorFlow 2.0 supports distributed training and multi-GPU training. Python development with eager execution is recommended, even though the old Session-based model is still supported.[26] Moreover, many APIs have been replaced or renamed. Code from TensorFlow 1 can be easily converted to TensorFlow 2.0 using an automatic conversion script.

TensorFlow's practical applications are image object detection, classification, language translation, voice recognition, text analysis, and many more. TensorFlow is used in many Google products such as Search, Gmail, and Translate. However, many other companies such as Intel, Coca-Cola, Airbus, Airbnb, or China Mobile also use TensorFlow.

4.2 KERAS

Keras is a high-level API for NNs. Keras promises to offer a modular, user-friendly, and extensible API that allows users to build, train, and utilize neural networks quickly and efficiently. It is written in Python and can use TensorFlow, Theano, and CNTK as backends. TensorFlow is the default, but it may be changed to any of the other options. The modularity principle means that models consist of combinable stand-alone modules. Stand-alone modules are neural layers, cost functions, activation functions, optimizers, initialization schemes, and regularizations. Such modules can be combined to create new models.

Moreover, it is possible to develop new custom modules. Keras provides two ways to define models: the functional API and the sequential model.[27] The former allows the definition of more complex models such as multi-output models, directed acyclic graphs, or models with shared layers. In contrast, a list of layer instances defines the sequential models. The HDF5 file format, which is unique to Keras, may be used to export trained models. Keras models may be distributed to various platforms, including Android using TensorFlow, iOS using Apple's, CoreML, Raspberry Pi, browsers using Keras.js or WebDNN, and Google Cloud.

4.3 MOBILE NET

MobileNet is a Convolutional Neural Network architecture developed the Google employees Howard et al. and was published in the paper “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications” in 2017. As the name suggests, it is optimized for mobile devices.[28] For making the model smaller and more efficient, MobileNet makes use of depthwise separable convolutions. Such convolutions factorize ordinary convolutions into a depthwise convolution and a pointwise convolution, a 1x1 convolution. Filters are separately applied to each channel by the depthwise convolution. Afterward, the outputs of the depthwise convolution are combined using the pointwise convolution. This way, filtering and the combination are separated into two layers, whereas ordinary convolutions do this in one layer.[29] The convolution is factorized, which reduces model size and the number of computations. MobileNet is made up of 28 layers in total. The first layer is a convolutional layer in the traditional sense. There are 13 pairs of depthwise and pointwise convolutional layers after that. The network architecture of MobileNet is given in the table below (See table 4.3.1).[28] At the end are one average polling, one fully connected, and one SoftMax layer. Batch normalization and ReLU follow all layers, except the fully connected layer. The computational costs and the size of the NN can be further influenced using the parameters α and ρ. α is called width multiplier and can be used to make the NN thinner.[30] ρ is called resolution multiplier with corresponds to the image input resolution. MobileNet is intended for computer vision tasks such as image recognition or object detection. On the ImageNet data a MobileNet with α = 1.0 and ρ = 224 reaches an accuracy of 70.6%.

In 2018 Sandler et al. published a new version in the paper "MobileNetV2: Inverted Residuals and Linear Bottlenecks". The MobileNetV2 architecture also makes use of depthwise convolutional layers. However, it introduces new building blocks called bottleneck residual blocks. These blocks comprise a 1x1 pointwise convolutional layer, a 3x3 depthwise convolution, and finally, a 1x1 pointwise convolutional layer.[28] After the first two layers, ReLU6 is applied, whereas the last layer has no non-linearity. Removing the non-linearity further improves the performance, according to the authors. The architecture has at the beginning a standard convolutional layer. After that, follow 19 residual bottleneck blocks. Some bottlenecks are grouped. The first block in each group has a one-stride stride, while the second has a two-stride stride. One 1x1 pointwise convolutional layer, one 7x7 average-pooling layer, and another 1x1 pointwise convolutional layer follow the bottleneck blocks. On ImageNet, MobileNetV3 achieves an accuracy of 72.0 percent, which is greater than MobileNetV2. At the same time, it is smaller with 3.4M parameters compared to 4.2M of the first generation (α = 1.0 and ρ = 224 for both models).

Table 4.3.1: Network Architecture of MobileNet

Input	Operator	Sides	Channel	Exp size
224 x 224 x 3	Conv2D, 3x3	1	16	-
224 x 224 x 16	Bottleneck, 3 x 3	2	16	16
112 x 112 x 16	Bottleneck, 3 x 3	2	24	72
56 x 56 x 24	Bottleneck, 5 x 5	2	40	96
28 x 28 x 40	Bottleneck, 5 x 5	1	40	240
28 x 28 x 40	Bottleneck, 5 x 5	1	40	240
28 x 28 x 40	Bottleneck, 5 x 5	1	48	120
28 x 28 x 48	Bottleneck, 5 x 5	2	96	288
14 x 14 x 96	Bottleneck, 5 x 5	1	96	576
14 x 14 x 96	Conv2D, 1 x 1	1	576	-
14 x 14 x 576	Avg Pooling2D	1	-	-
1 x 1 x 576	Conv2D, 1 x 1	1	1280	-
1 x 1 x 1280	Conv2D, 1 x 1	1	N = 2	-

4.4 UNITY

Unity Technologies is a game creation engine for 3D and 2D games. Games can be exported to various platforms, including Windows, Linux, Mac OS desktop PCs, Android and iOS smart devices, and game consoles such as the Nintendo Switch, Xbox, and PlayStation. Moreover, Unity also has support for VR and MR.[31] The HoloLens is also officially supported through UWP apps. The game logic is programmed in C#. Therefore, many C# libraries can be used inside Unity games. Countless successful games such as Hearthstone, Monument Valley 2, or Cities: Skylines were developed using Unity. However, the engine can also be used for professional applications like the visualization of technical models.[32]

CHAPTER 5: IMPLEMENTATION

This chapter describes the application's design using unity 3D, developing, and implementing a trained model using deep learning. The better version of the MobileNet model is chosen and prepared using our dataset. The model is turned into a frozen graph and loaded into the framework to execute in an application. The implementation procedure is listed below as a conceptual diagram; This includes the integration of the runtime augmented reality application with the deep learning model. The author had a brief description of the development of the application in the upcoming subheading.

5.1 METHODOLOGY

This section discusses the integration of two different technologies and efficiency. Firstly, MobileNets, a deep convolutional neural network, is adopted as a lightweight deep learning model and runs efficiently in a mobile device with high accuracy. Secondly, augmented reality technology is integrated to display the bounding box detected from the result generated from images. The deep learning and augmented reality technology are integrated using the TensorFlow library files within a unity framework. These two technologies perform together in a runtime application on a mobile device. The proposed framework is summarized in Figure 5.1.1

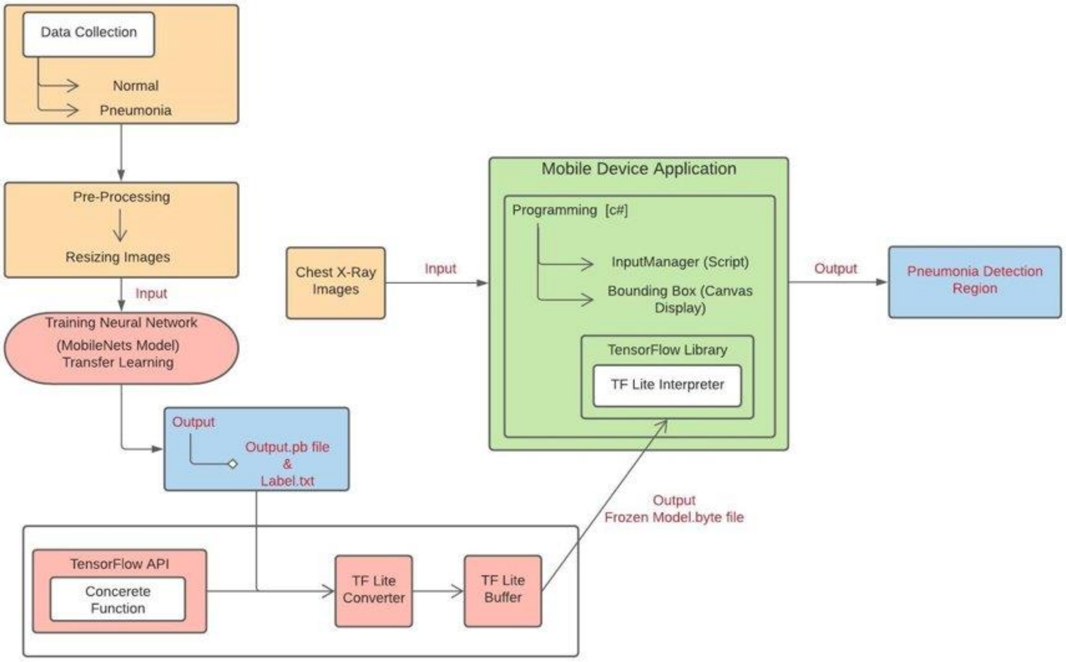


Figure 5.1.1: Conceptual diagram of the proposed framework

5.2 DATASET DESCRIPTIONS

The input data used in this paper is inherited from the Kaggle Pneumonia Challenge chest x-ray images. It consists of 9418 chest x-ray images and labels in CSV file format with the ground truth value, and the dataset is split into three portions: Train, Test, and Validate. Each part has two subdivisions, which are Pneumonia and Normal. The sample images of pneumonia and normal are shown in the table 5.2.1 below. This chest x-ray examination is beneficial.[33] It allows doctors or radiologists to analyze internal diseases such as blood vessels, lungs, and abnormal spots; pneumonia, which is caused due to infection in the lungs, can be detected from a chest x-ray. This research developed automated models to see pneumonia from chest x-ray graphs without any human assistance. These chest x-ray images are analyzed and screened for quality control by a chest radiologist, which helped remove all low-quality or unreadable scans. The dataset which we use to train the model is a chest x-ray of both genders. Annotation value is provided as Bounding

Box localization labeled as, and from the center of the bounding box, (W) as width and (H) as height (See figure 5.2.1). The details about the portions of the images are mentioned in the table below (See table 5.2.1).

Table 5.2.1: Details about the portions of images from the dataset

	Pneumonia	Normal
Training Dataset	4622	4314
Testing Dataset	232	234
Validation Dataset	8	8

The primary goal of using a Deep Convolutional Neural Network in most image classification and detection tasks is to reduce the model's computational complexity and increase if the input is images. The size of the pictures is resized from 1024 x 1024 to 224 x 224. This step is programmed in python to automatically resize all training images in folders instead of manually resizing each image on separate software, eradicating tediousness.[9] This step was done to produce an excellent and reliable output while maintaining minimum possible parameters such as time, processing hardware requirements, energy consumption, and memory.

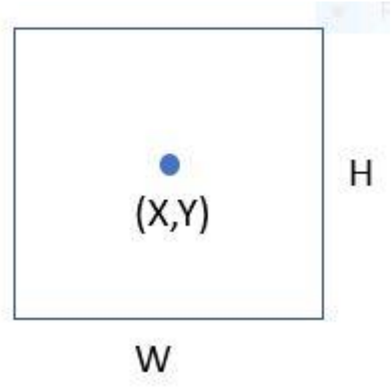


Figure 5.2.1: Bounding Box plot values of the given dataset

5.3 ARCHITECTURE

MobileNets is a lightweight deep learning model developed to support and perform mobile and embedded vision applications. The core layer of the MobileNets network structure consists of depthwise separable filters; the layer of this MobileNets consists of factorized convolution, standard convolution into a depthwise convolution, and convolution, which is also called a pointwise convolution. A single filter is applied to each filter in the depthwise convolution. The network architecture of the MobileNet is given in (table 4.3.1). The pointwise convolution uses 1x1 convolution to join the depthwise convolution outputs. This research uses MobileNets V3 to detect pneumonia and embed it in head mounted devices.[31] MobileNetV3 has been adjusted and developed for mobile phone CPU the combination of hardware-aware network architecture search (NAS) and Net Adapt algorithm. It was subsequently improved through novel architectural improvements.

As observed, more layers are incorporated in MobileNets V3, thus increasing the network's reliability and accuracy. The layers of the model are shown in the table below. For this pre-trained model, convolutional layers are the ones responsible for extracting features. There is no specific feature (color, edge, and the like) to be removed, unlike previous machine learning techniques; for MobileNets V3, the convolutional layers will define the quality of the training dataset. The features may be one, a combination of some, or complex ones. The following table lists the architecture of the pre-training model used. There are various parameters shown per operator and layer of the architecture. However, the focus of the delivery table is the bottleneck layer. Bottleneck layers maintain the "deepness" of the network. Thus, a Deep learning approach while reducing the number of layers per stage. This model is compared with other models that use large dimensions for layers, which consume much time and memory usage.

The objective of the model is to distinguish a healthy lung from the infected one by learning. There are two categories with labels: NORMAL and PNEUMONIA. All the resized images are taken as input for training the model by reiterating transfer learning. Based on the performance, the model's expected output is bounding box values of the detected region, which are AR applications' input. In this study, the model (.pb file) and labels.txt file containing the labels such as normal and pneumonia were developed.

5.4 INTEGRATION

The Unity framework was used to create the application, including C# on the backend and User Interface components for the front-end framework. The only needed file is the constructed frozen graph of the convolutional network to integrate the deep learning model with the augmented reality (AR) applications.[25] The frozen graph is also relevant to correctly set the model. This frozen graph is a file that saves the graph and weights of a trained model as a single file. The construction of a frozen graph is built using the TensorFlow Lite converter tool. This tool converts the model (.pb) file into a frozen graph (.byte) file to perform into AR applications. The model's detected regions bounding box values are taken as input to create and overlay the application's augmented bounding box. Several C# manuscripts are made to send and retrieve images frame per second from model to AR application.

CHAPTER 6: IMPLEMENTATION & EXPERIMENT AND EVALUATION

The following chapter describes the experiment execution and evaluation with their results.

6.1 EXPERIMENT

In this section, MobileNets V3, a pre-trained on ImageNet dataset, is fine-tuned, trained, and tested on normal and pneumonia chest x-ray images. The dataset consists of 8936 images for training and 464 images for testing the models. All the training images are resized during the pre-processing. The Python platform is used to code using TensorFlow to train the model, with a total number of epochs of 96, a learning rate of 0.00001, and an 8-batch size. The table below represents the system configuration used while training the model.

Table 6.1.1: System Configuration

Version	Python	cuDNN	CUDA
Tensorflow_gpu-1.4.0	2.1, 3.3 – 3.6	6	8

The training method is the transfer learning method, which is not new, either in the general case of image detection or even more novel techniques. This block's output was further flattened and given to the fully connected layers, which used Conv2D, Bottleneck, and AvgPooling2D in the MobileNets V3 model. The training was done by changing the last few layers. The weights initially loaded for the convolutional layer have not been used and have not been changed. The selection of MobileNets V3 as an on-device classifier based on DCNN is most favorable. It is an open-source software from TensorFlow that works with minimal hardware and software requirements met with smartphones available today. To summarize, pneumonia's confidence level is based on two detection labels defined as 'Normal' or 'Pneumonia.'

After the training process is finished, the trained model is converted to the lite version used for mobile devices using the TensorFlow Lite Converter and can be used as a quantized or float model. Compared with the floating model on mobile devices, the quantized Lite model shows detection speed four times faster. The mobile application uses the TensorFlow Lite API to interact with the quantization model by providing resized image frames received from the mobile camera. The model feeds the received image to the trained model and returns a set of bounding box values. Every detection consists of a bounding rectangle of the corresponding region in the image, the classification value, and the given label name's index in the list of class names.

6.2 TENSORFLOW LIBRARY

TensorFlow Sharp is an excellent runtime library file that imports, loads, runs, and delivers output from the model to other script files. TensorFlow library file creates a Unity platform environment where the Low-Level TensorFlow API (This API is Hard and Requires more setup). This Library file Understands the model and runs through the runtime of the application using the following script. The Namespaces used in the script are represented in the image and mentioned in the appendix (See figure 6.2.1, 6.2.2).

6.3 BOUNDING BOX TEXTURE REGION

Texture Tools is a script that helps crop and project the virtual bounding box around the detected region. It helps notify the radiologist or the medical expert about Pneumonia and its region value within the User Interface. The script CatalogUtil and CatalogItem help to display the label of the detected object within the bounding box. The namespace and function of script is represented in the appendix (See figure 6.3.1). The only needed file is the constructed frozen graph of the convolutional network to create the application. However, before applying the application, it must be determined if the network can accurately categorize and detect capable of accurately categorizing and detecting Pneumonia on chest x-ray pictures, regardless of whether the image is labeled as PNEUMONIA or NORMAL. To do so, the network uses test pictures (which may or may not be part of the original training data set) to check whether the model is effective in classification and detection.

In the section on computational findings, the accuracy and region detection results are addressed. If the required performance is reached, the Android application is integrated into the Unity framework, which is where the Mobile application is created. On the canvas Game Object, the image collected from the camera will be shown (Unity feature). This Game Object element contains a raw picture, which is a built-in feature of every camera program. The model will be provided for each image for detection. On the Canvas Game Object, later detection and localization will be presented.

6.4 EVALUATION AND RESULTS

The trained model is evaluated using the intersection-over-union method. To measure the performance of the model, we not only used the evaluation indicators to calculate the accuracy and the false positive rate (FPR) under varying intersection over union (IoU) thresholds and calculated the precision and recall rate at a single IoU threshold. We simply write the IoU threshold as T(IoU). This helps to evaluate if the model is reliable; when higher the confidence of the predicted bounding box, the more likely it is to hit the actual pneumonia area. The IoU of a set of predicted bounding boxes and GT bounding boxes is calculated as follows (Equation 6.4.1):

Equation 6.4.1 $$IoU\ (A,B) = A \cap B / A \cup B$$

Where A and B denote the area enclosed by the predicted bounding box and GT bounding box, respectively, we set five different T(IoU) from 0.3 to 0.7 to find out whether the target has been hit or missed. For example, T(IoU), greater than 0.5, means a predicted bounding box region intersects over a union with a ground truth bounding box is more significant than 0.5. All four (TP/FP/TN/FN) are obtained for each threshold by comparing the predicted result with all GT objects. Therefore, the accuracy and recall rate can be calculated at the changing T (IoU) point, which can be written as:

Equation 6.4.2 $$Precision = \frac{TP}{TP+FP}$$

Equation 6.4.3 $$recall = \frac{TP}{TP+FN}$$

In our model, low precision and a high recall indicate that there are larger samples for model error detection and lesser leaks. Precisely, the T(IoU) is set from 0.4 to 0.6 with a step of 0.0045 according to the official metrics. And the average accuracy and FPR are calculated under this set of thresholds. The calculated accuracy and FPR for our metrics are 0.278 and 0.198. The figure shown below is the runtime application result, captures while detecting the pneumonia region and augmenting the bounding box on the detected region (See Figure 6.4.1).

Table 6.4.1: The precision and recall table

Method	**Indicator**	**T(IoU)**			
		0.4	0.5	0.6	0.7
Our Metrics	Precision	0.629	0.499	0.365	0.389
	Recall	0.818	0.816	0.770	0.798

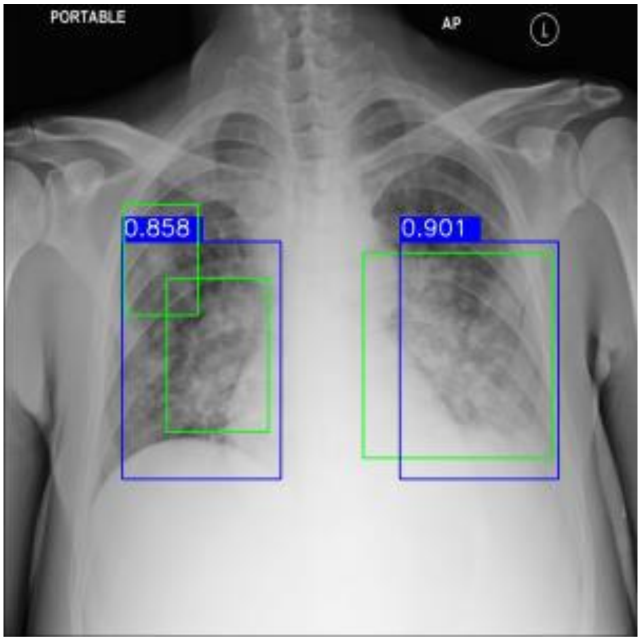
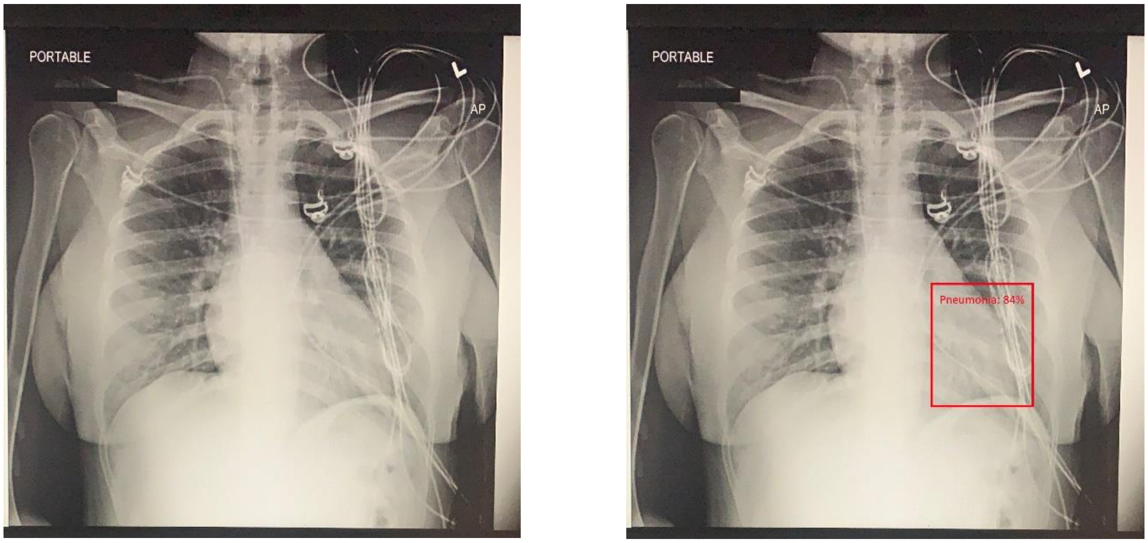


Figure 6.4.1: The green and blue boxes in the figure are bounding box markers for ground truth and prediction, respectively. The numbers in the figure represent the confidence level of the predicted bounding box



A. B.

Figure 6.4.2: Bounding Box region and pneumonia detection in a runtime application on HoloLens emulator. A) placing chest X-ray image Infront of runtime camera B) One of the images per frame detects pneumonia and augmented Bounding Box on the localized region

CHAPTER 7: CONCLUSION

This thesis concludes by developing a runtime application to visualize pneumonia in chest x-ray images by integrating two different platforms, deep learning and augmented reality. A pre-trained model is employed and trained using the transfer learning method. The training is done using a 2070 RTX and SSD memory system, allowing faster training and better data use on the unity framework. Medical practitioners usually use chest x-ray to diagnose this phenomenon, greatly influenced by their visual inspection perception. Accurate detection and visualization of pneumonia play a vital role here. Obtaining the patient's chest x-ray, the practitioners or radiologists can use this application to get the instant result of classified pneumonia and its localization information. They are placing the chest x-ray image in front of the head-mounted device aids to capture the image and detect pneumonia frame per second, which supports them in visualizing and improving the prediction accuracy to proceed with the diagnosis process. The application is meticulously performed with the delayed result to identify pneumonia; however, the application is made in real-time, the detection may not display on the app. Future studies will take these flaws into account. Furthermore, since the HoloLens gen one has terminated, the program may or may not operate on the HoloLens gen two devices.

REFERENCES

[1] G. Labhane, R. Pansare, S. Maheshwari, R. Tiwari, and A. Shukla, "Detection of Pediatric Pneumonia from Chest X-Ray Images using CNN and Transfer Learning," *Proc. 3rd Int. Conf. Emerg. Technol. Comput. Eng. Mach. Learn. Internet Things, ICETCE 2020*, no. February, pp. 85–92, 2020, doi: 10.1109/ICETCE48199.2020.9091755.

[2] A. Chaudhary, A. Hazra, and P. Chaudhary, "Diagnosis of Chest Diseases in X-Ray images using Deep Convolutional Neural Network," *2019 10th Int. Conf. Comput. Commun. Netw. Technol. ICCCNT 2019*, pp. 6–11, 2019, doi: 10.1109/ICCCNT45670.2019.8944762.

[3] G. Bras, V. Fernandes, A. C. De Paiva, G. B. Junior, and L. Rivero, "Transfer Learning Method Evaluation for Automatic Pediatric Chest X-Ray Image Segmentation," *Int. Conf. Syst. Signals, Image Process.*, vol. 2020-July, pp. 128–133, 2020, doi: 10.1109/IWSSIP48289.2020.9145401.

[4] "COMPUTER ASSISTED READING OF CHEST RADIOGRAPHS Nandinee Fariah Haq , Z . Jane Wang The University of British Columbia," pp. 20–24, 2019.

[5] G. IEEE Engineering in Medicine and Biology Society. Annual International Conference (41st : 2019 : Berlin, IEEE Engineering in Medicine and Biology Society, and Institute of Electrical and Electronics Engineers, *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC) : Biomedical Engineering Ranging from Wellness to Intensive Care : 41st EMB Conference 2019 : July 23-27, Berlin.* .

[6] B. Kelly, "The chest radiograph," *Ulster Med. J.*, vol. 81, no. 3, pp. 143–148, 2012, doi: 10.1016/b978-0-323-39952-4.00004-4.

[7] D. K. Iakovidis and E. Papageorgiou, “Intuitionistic fuzzy cognitive maps for medical decision making,” *IEEE Trans. Inf. Technol. Biomed.*, vol. 15, no. 1, pp. 100–107, 2011, doi: 10.1109/TITB.2010.2093603.

[8] X. Images, “Quality of Chest X-ray Images Using,” vol. 00, pp. 931–934, 2005.

[9] L. Luo *et al.*, “Deep Mining External Imperfect Data for Chest X-Ray Disease Screening,” *IEEE Trans. Med. Imaging*, vol. 39, no. 11, pp. 3583–3594, 2020, doi: 10.1109/TMI.2020.3000949.

[10] S. I. Astuti, S. P. Arso, and P. A. Wigati, “Machine learning based common radiologist level pneumonia detection on chest X-rays,” *Anal. Standar Pelayanan Minimal Pada Instal. Rawat Jalan di RSUD Kota Semarang*, vol. 3, pp. 103–111, 2015.

[11] H. Mane, P. Ghorpade, and V. Bahel, “Computational Intelligence Based Model Detection of Disease using Chest Radiographs,” *Int. Conf. Emerg. Trends Inf. Technol. Eng. ic-ETITE 2020*, pp. 1–5, 2020, doi: 10.1109/ic-ETITE47903.2020.484.

[12] D. Varshni, K. Thakral, L. Agarwal, R. Nijhawan, and A. Mittal, “Pneumonia Detection Using CNN based Feature Extraction,” *Proc. 2019 3rd IEEE Int. Conf. Electr. Comput. Commun. Technol. ICECCT 2019*, pp. 1–7, 2019, doi: 10.1109/ICECCT.2019.8869364.

[13] V. Bhagat and S. Bhaumik, “Data Augmentation using Generative Adversarial Networks for Pneumonia classification in chest Xrays,” *Proc. IEEE Int. Conf. Image Inf. Process.*, vol. 2019-Novem, pp. 574–579, 2019, doi: 10.1109/ICIIP47207.2019.8985892.

[14] E. Ayan and H. M. Ünver, “Diagnosis of pneumonia from chest X-ray images using deep learning,” *2019 Sci. Meet. Electr. Biomed. Eng. Comput. Sci. EBBT 2019*, pp. 0–4, 2019, doi: 10.1109/EBBT.2019.8741582.

[15] J. Eichel, “E Fficient N Etwork S Tructure,” no. 2014, pp. 2016–2019, 2018, [Online].

Available: https://openreview.net/pdf?id=BkgXT24tDS.

[16] C. Zhang and P. C. Woodland, “DNN speaker adaptation using parameterised sigmoid and ReLU hidden activation functions,” *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc.*, vol. 2016-May, no. 1, pp. 5300–5304, 2016, doi: 10.1109/ICASSP.2016.7472689.

[17] T. Hossen, A. S. Nair, R. A. Chinnathambi, and P. Ranganathan, “Residential Load Forecasting Using Deep Neural Networks (DNN),” *2018 North Am. Power Symp. NAPS 2018*, 2019, doi: 10.1109/NAPS.2018.8600549.

[18] H. Sharma, J. S. Jain, P. Bansal, and S. Gupta, “Feature extraction and classification of chest X-ray images using CNN to detect pneumonia,” *Proc. Conflu. 2020 - 10th Int. Conf. Cloud Comput. Data Sci. Eng.*, pp. 227–231, 2020, doi: 10.1109/Confluence47617.2020.9057809.

[19] J. R. Ferreira, D. Armando Cardona Cardenas, R. A. Moreno, M. De Fatima De Sa Rebelo, J. E. Krieger, and M. Antonio Gutierrez, “Multi-View Ensemble Convolutional Neural Network to Improve Classification of Pneumonia in Low Contrast Chest X-Ray Images,” *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS*, vol. 2020-July, pp. 1238–1241, 2020, doi: 10.1109/EMBC44109.2020.9176517.

[20] Z. Zhang, D. Weng, H. Jiang, Y. Liu, and Y. Wang, “Inverse Augmented Reality: A Virtual Agent’s Perspective,” *Adjun. Proc. - 2018 IEEE Int. Symp. Mix. Augment. Reality, ISMAR-Adjunct 2018*, pp. 154–157, 2018, doi: 10.1109/ISMAR-Adjunct.2018.00056.

[21] U. Rehman and S. Cao, “Augmented Reality-Based Indoor Navigation Using Google Glass as a Wearable Head-Mounted Display,” *Proc. - 2015 IEEE Int. Conf. Syst. Man, Cybern. SMC 2015*, pp. 1452–1457, 2016, doi: 10.1109/SMC.2015.257.

[22] H. Chung, “Development of a Head-Mounted Mixed Reality Museum Navigation System,” pp. 111–114, 2021.

[23] Z. Zhang, B. Cao, D. Weng, Y. Liu, Y. Wang, and H. Huang, “Evaluation of Hand-Based Interaction for Near-Field Mixed Reality with Optical See-Through Head-Mounted Displays,” *25th IEEE Conf. Virtual Real. 3D User Interfaces, VR 2018 - Proc.*, pp. 739–740, 2018, doi: 10.1109/VR.2018.8446129.

[24] 4 Experience, “HoloLens2 VS HoloLens1,” 2020, [Online]. Available: https://4experience.co/hololens-2-vs-hololens-1-whats-new/.

[25] H. Subakti and J. R. Jiang, “Indoor Augmented Reality Using Deep Learning for Industry 4.0 Smart Factories,” *Proc. - Int. Comput. Softw. Appl. Conf.*, vol. 2, pp. 63–68, 2018, doi: 10.1109/COMPSAC.2018.10204.

[26] “TensorFLow 2.0,” 2021, [Online]. Available: https://www.tensorflow.org/guide/effective_tf2.

[27] M. Hu *et al.*, “Learning to recognize chest-xray images faster and more efficiently based on multi-kernel depthwise convolution,” *IEEE Access*, vol. 8, pp. 37265–37274, 2020, doi: 10.1109/ACCESS.2020.2974242.

[28] D. Sinha and M. El-Sharkawy, “Thin MobileNet: An Enhanced MobileNet Architecture,” *2019 IEEE 10th Annu. Ubiquitous Comput. Electron. Mob. Commun. Conf. UEMCON 2019*, pp. 0280–0285, 2019, doi: 10.1109/UEMCON47517.2019.8993089.

[29] H. Y. Chen and C. Y. Su, “An Enhanced Hybrid MobileNet,” *2018 9th Int. Conf. Aware. Sci. Technol. iCAST 2018*, pp. 308–312, 2018, doi: 10.1109/ICAwST.2018.8517177.

[30] M. M. Yapıcı and N. Topaloğlu, “Performance comparison of deep learning frameworks,” *Comput. Informatics*, vol. 1, no. 1, pp. 1–11, 2021, [Online]. Available:

https://dergipark.org.tr/tr/pub/ci.

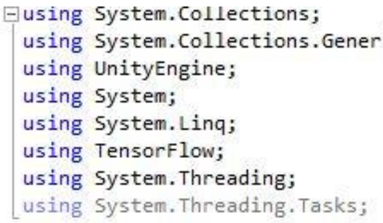
[31] E. Kucera, O. Haffner, and R. Leskovsky, “Interactive and virtual/mixed reality applications for mechatronics education developed in unity engine,” *Proc. 29th Int. Conf. Cybern. Informatics, K I 2018*, vol. 2018-January, pp. 1–5, 2018, doi: 10.1109/CYBERI.2018.8337533.

[32] M. J. Maasthi, H. L. Gururaj, V. Janhavi, K. Harshitha, and B. H. Swathi, “An Interactive Approach Deployed for Rhinoplasty using Mixed Reality,” *2020 Int. Conf. Commun. Syst. NETworkS, COMSNETS 2020*, pp. 680–682, 2020, doi: 10.1109/COMSNETS48256.2020.9027491.

[33] G. V. Sreena, N. Ponraj, and L. P. Deepa, “Study on public chest X-ray data sets for lung disease classification,” *2021 3rd Int. Conf. Signal Process. Commun. ICPSC 2021*, no. May, pp. 54–58, 2021, doi: 10.1109/ICSPC51351.2021.9451726.

APPENDIX

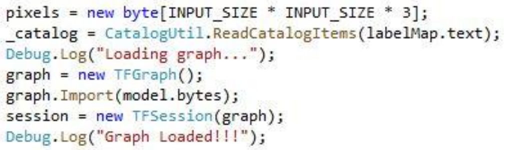
The given diagram represents the namespace available and in the TensorFlow library. These namespace carries numerous methods. Some of the functions are input/output, session to load the graph, etc., The figure XX below represents the namespace of given library.



```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using System;
using System.Linq;
using TensorFlow;
using System.Threading;
using System.Threading.Tasks;
```

Figure 6.2.1: Namespace of TensorFlow library file

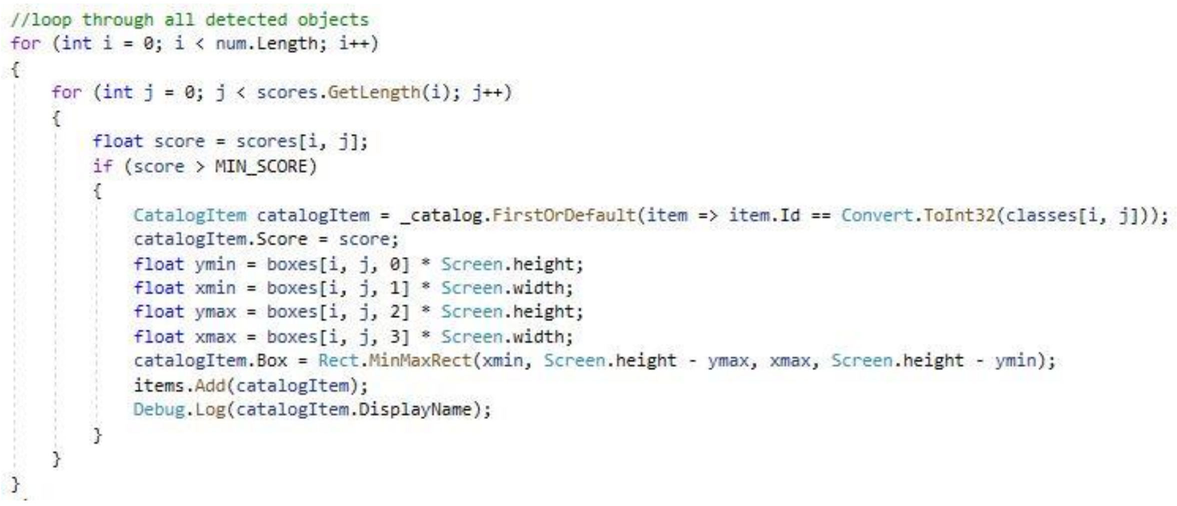
The must require variables and functions are utilized in the script which helps to integrate deep learning (frozen graph) and bounding box region script. The figure XX below loads the DL model in C# script.



```
pixels = new byte[INPUT_SIZE * INPUT_SIZE * 3];
_catalog = CatalogUtil.ReadCatalogItems(labelMap.text);
Debug.Log("Loading graph...");
graph = new TFGraph();
graph.Import(model.bytes);
session = new TFSession(graph);
Debug.Log("Graph Loaded!!!");
```

Figure 6.2.2: Model utilization in C# script

Various scripts are used to process the picture collected by the camera; specifics are given in section 6.3. These images are cropped and bounding box regions are applied to them. This procedure repeats itself. The researcher used a for loop to build a bounding box area on the image depending on the plot value (See figure 6.3.1).



```
//loop through all detected objects
for (int i = 0; i < num.Length; i++)
{
    for (int j = 0; j < scores.GetLength(i); j++)
    {
        float score = scores[i, j];
        if (score > MIN_SCORE)
        {
            CatalogItem catalogItem = _catalog.FirstOrDefault(item => item.Id == Convert.ToInt32(classes[i, j]));
            catalogItem.Score = score;
            float ymin = boxes[i, j, 0] * Screen.height;
            float xmin = boxes[i, j, 1] * Screen.width;
            float ymax = boxes[i, j, 2] * Screen.height;
            float xmax = boxes[i, j, 3] * Screen.width;
            catalogItem.Box = Rect.MinMaxRect(xmin, Screen.height - ymax, xmax, Screen.height - ymin);
            items.Add(catalogItem);
            Debug.Log(catalogItem.DisplayName);
        }
    }
}
```

Figure 6.3.1: Bounding Box Script Line

VITA

Jeevarathinam Senthilkumar was born in the Tamil Nadu city of Dharmapuri. He earned a bachelor's degree in information technology from Bannari Amman Institute of Technology in 2016. He was recruited as a web developer by a firm called "Bacelor.inc" after graduation. He needs to create a web-based game for a variety of clientele. He intended to update his carrier since he was acquainted with the Unity gaming engine. At the University of Texas at El Paso, he began a Master of Science in Systems Engineering program. He also worked as a research assistant for an Industrial, Manufacturing, and Systems Engineering chair professor. His profession incorporates augmented reality and virtual reality, and he has had the opportunity to try out future equipment such as the Microsoft HoloLens gen one and the HTC Vive. He also obtained familiarity with both technologies by designing an application. He created materials using his superior knowledge of these technologies and the help of his professor. For the Industrial, Manufacturing, and Systems Engineering Department, he created a new course called Augmented and Virtual Reality. In collaboration with Drexel University, he and his professor worked on a project named DoED MSEIP. He was also active in student groups, serving as president of the Indian Student Association (ISA). He earned a 3.9 GPA while earning his master's degree in Systems Engineering in the fall of 2021.

Contact Information: sjeevarathinam29@gmail.com