Medical Diagnostic Assistance with Deep Learning

Project Report

Submitted by:

Taha Zanzibarwala (1641057)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

INFORMATION AND COMMUNICATION TECHNOLOGY

(ICT)

at

Ahmedabad University

School of Engineering and Applied Sciences (SEAS)

Ahmedabad, Gujarat

April, 2020

DECLARATION

I hereby declare that the project entitled "Medical Diagnostic Assistance with Deep Learning" submitted for the B. Tech. (ICT) degree is my original work and the project has not formed the basis for the award of any other degree, diploma, fellowship or any other similar titles.

[illegible]

Taha Zanzibarwala

Place: Ahmedabad, India

Date: 15th April, 2020

CERTIFICATE

This is to certify that the project titled “Medical Diagnostic Assistance with Deep Learning” is the bona fide work carried out by Taha Zanzibarwala, a student of B Tech (ICT) of School of Engineering and Applied Sciences at Ahmedabad University during the academic year 2019-2020, in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology (Information and Communication Technology) and that the project has not formed the basis for the award previously of any other degree, diploma, fellowship or any other similar title.

Dr. Sanjay Chaudhary, Interim Dean and Professor,
School of Engineering and Applied Science,
Dean of Students, Ahmedabad University

Place: Ahmedabad, India

Date: 15th April, 2020

ABSTRACT

Rural and remote areas of many developing countries face obstructions in timely and accurate medical assistance, resulting in slow and obscure diagnosis and therefore treatment. Medical professionals also sometimes are inexperienced in some domains and need further assistance. The proposed system aims to augment the diagnostic skills of professionals and classify an abnormality using only image data.

The system will use image inputs (RGB images or XRays) and no further information to generate a prediction. Deep Learning models such as CNNs will be used for feature extraction of the image data for useful analysis. The system is planned to be deployed on various platforms for ease of use and wide reach.

Given the scarce research and development in diagnostic AI, combined with the weak confidence of society on an algorithm to diagnose their ailments, state of the art and benchmarks are varied and decentralized. Furthermore, medical data is key to someone's privacy and therefore is not readily available. These factors sum up in the accuracy for state of the art to be modest. Our proposed systems reach in the vicinity of these benchmarks although with different classes and different training methods. As discussed ahead, the results can be improved in future scope.

ACKNOWLEDGMENT

I would like to thank Mr. Jigarkumar Mori, CEO and Director, Geeky Bee AI Pvt Ltd. for providing me the opportunity to work with the team and providing his assistance and leadership throughout my project.

Secondly, I would also like to thank Dr. Sanjay Chaudhary, Ahmedabad University for guiding me throughout my academic tenure and providing his assistance throughout this project.

I would also like to extend my thanks to my team at Geeky Bee AI for helping me learn and implement better at every step. Furthermore, I would like to thank my parents, friends and faculty members at the School of Engineering and Applied Science, Ahmedabad University who supported me through my learning and adventurous journey. Their support will always be cherished.

Taha Zanzibarwala

LIST OF FIGURES

GANTT CHART

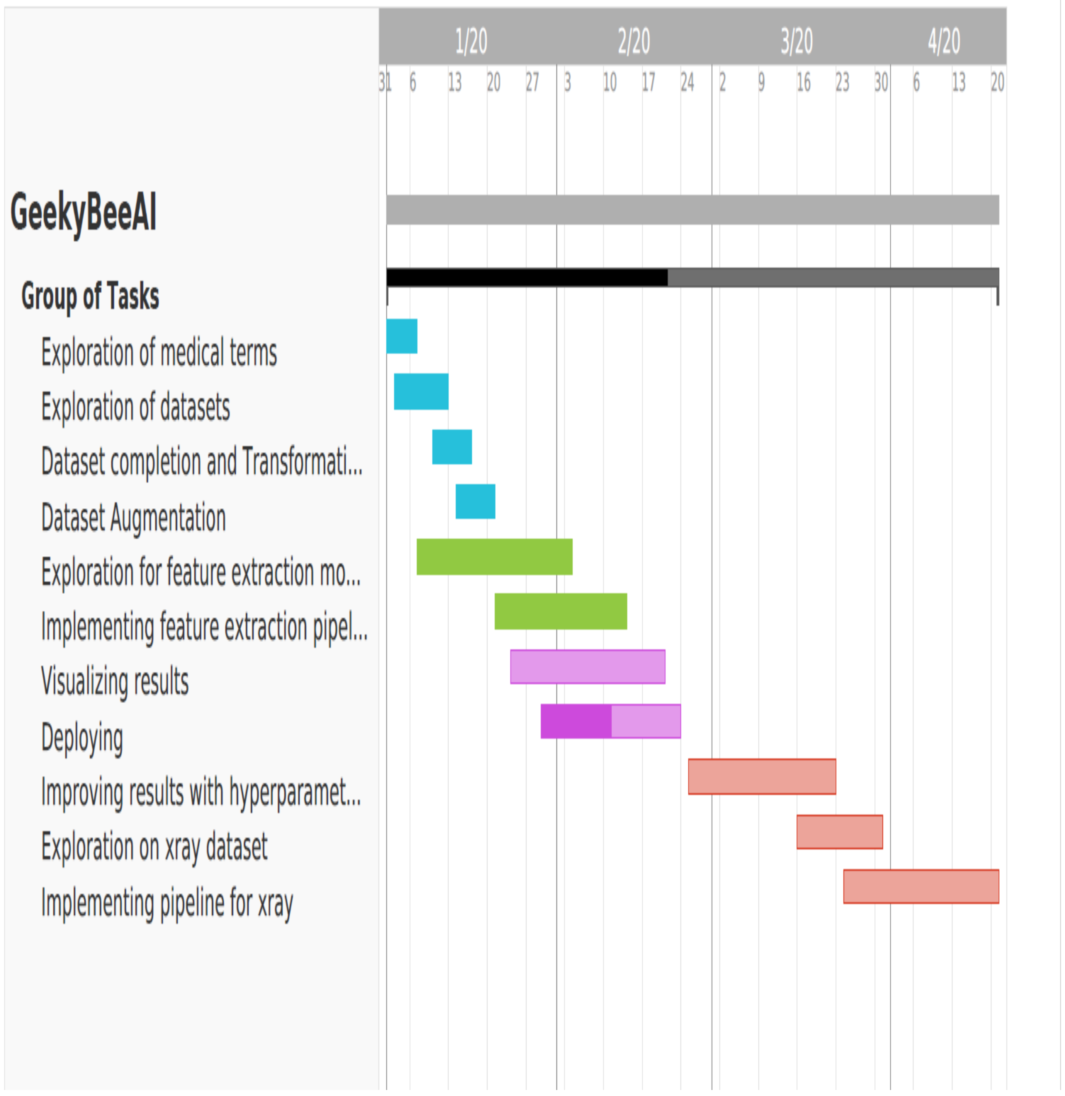

1/20
2/20
3/20
4/20
31
6
13
20
27
3
10
17
24
2
9
16
23
30
6
13
20
GeekyBeeAI
Group of Tasks
Exploration of medical terms
Exploration of datasets
Dataset completion and Transformati...
Dataset Augmentation
Exploration for feature extraction mo...
Implementing feature extraction pipel...
Visualizing results
Deploying
Improving results with hyperparamet...
Exploration on xray dataset
Implementing pipeline for xray

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Problem Definition

Timely and accurate diagnosis of ailments, preferably in their benign stages has a huge impact on the treatment and recovery of patients. But, rural and remote areas face scarcity of proper medical facilities or professionals leading to late or improper diagnosis of their ailments. In many cases, a certain degree of difference is also observed in the diagnosis of doctors. In light of these issues, we aim to augment the skills of the medical professionals using the tools of Computer Vision and Deep Learning to provide accurate and timely treatment to its users by classifying/predicting the ailment by image inputs.

1.2 Project Overview

The project revolves around developing an independent system augmenting the skills of medical professionals to classify a disease by image input. The project spanned over numerous domains of medical diagnosis, being femur detection, skin disease classification, chest X-Ray analysis and more. The proposed project work, currently, consists of two modules. Firstly a skin disease diagnosis model and secondly a chest x-ray diagnosis model.

The general goals or objectives for the same are described as below:

- To train a deep learning classifier on open source or private dataset for selected classes for both the modules.
- Extract features from the given input or saved on the local machine to classify the image as one of the classes.
- Deploy modules on various platforms for comparison to select the best working deployment.

The selected classes for the skin module are:

- Acne
- Scabies

- Psoriasis
- Vitiligo
- Warts
- Fungal Infection
- Leprosy

The selected classes for Xray module are:

- Effusion
- Pneumonia
- Pneumothorax

1.3 Hardware Specification

- Local Machine
  - HP Inspiron
  - RAM 16GB
  - Processor Intel(R) Core(TM) i7-7500U CPU
  - Storage 2 TB
  - OS Windows 10
  - Usage: Communication with the team, writing python scripts for training and preprocessing exploration
- Remote Machine 1
  - OS Ubuntu 18.04
  - Processor Intel(R) Core(TM) i7-7500U CPU
  - RAM 32GB
  - Usage: Computational analysis
- Remote Machine 2
  - OS Ubuntu 18.04
  - Processor Nvidia GTX 1080 Ti
  - RAM 12GB
  - Usage: Computational analysis

1.4 Software Specification

Software/Library used are:

- PyCharm IDE
- Google Colab
- Keras
- PyTorch
- OpenCV
- Matplotlib

2. LITERATURE SURVEY

2.1 Existing System

2.1.1 Skin Disease Diagnosis

Considering the selected classes for diagnosis, a similar classification system was not to be found. Furthermore, the medical records being highly anonymized and prohibited for commercial use, research in skin disease classification is considered in its nascent stage. One of the reliable datasets used for the module, The HAM10000 dataset [1] consisting of around 10000 labeled images of skin lesions. As the proposed system was to be lightweight and easily deployed on independent sources, MobileNet [2] was selected for the initial prototype. The [2] architecture is based on depthwise separable convolution which highly optimizes the CNN calculations and hence making it a highly efficient architecture. After certain discussions, the dataset [1] was deemed unfit for the problem statement and hence a custom dataset sourced from various databases was made available. On a custom dataset consisting of four classes of skin abnormalities, [3] scores high on the classification using image texture and color.

2.1.2 Lung Disease Diagnosis

Chest Xray dataset from the NIH Clinical Center [4] is a highly structured chest x-ray database of around 30,000 patients consisting of around 100,000 anonymized x-ray images. The database labels the images in 15 classes, of which 3 classes, highly diagnosed, were selected. The current state of the art for the dataset includes CheXNet [6]. The proposed model for [6] is a 121-layer densely connected neural network. Heatmaps were also induced to visualize regions where the abnormality of the x-ray is identified. Densely Connected Convolutional Networks [7] are efficient to train as the overall trainable parameters are decreased many-fold. Considering the previous aim to deploy the model on low-performance platforms, [7] constitutes as a viable option to train our data on.

2.2 Proposed System

The proposed system of the above-mentioned problem is to develop an image classification pipeline consisting mainly of a deep convolutional neural network for feature extraction and classification. Furthermore, to train the said classifier, the data pipeline will preprocess the images, be it camera images or x-rays, sending it to the selected classification model for training. The data collection process is explained in the respective systems. The data pipeline will also include data augmentation to increase class representation and reduce bias in the data.

2.2.1 Skin Disease Diagnosis

The custom dataset for the skin ailments was scraped from the internet, taking care that the images came under a public license. These images were highly mislabeled and contained various images not fit for training. Noisy and unusable images were removed. Deep learning models are data-hungry and therefore, our scraped data was inadequate. Dermnet [9] provides skin aliment images. These images were merged to form a complete dataset. As a feature extraction and classification model, VGG16 [8] was deemed the best fit. The depth of the model allowed us to store enough information for the problem. Furthermore, pre-trained weights for the model allowed us to save important computational time in training. The convolutional architecture has uniform 3x3 filters. These being the smallest receptive fields for the detection of changes, it gives an advantage over other networks. Originally the classifier consists of a fully connected neural network of 4096 channels followed by 1000 channels. The classifier was modified for further improving accuracy as 4096, 4096, 512, 7 channels respectively. The total layers being 16, with 13 convolutional layers and 3 fully-connected layers. As mentioned before, to speed up training pre-trained weights are used initially so the parameters are developed on ImageNet challenge. This helps in saving the initial parameter learning time which might be redundant for our dataset.

2.2.2 Lung Disease Diagnosis

The NIH [4] dataset selected for the problem had a high bias for certain classes and many images were labeled for multiple classes. Training our model for the mentioned problem, many of the x-ray images were to be filtered to fulfill the needs. Augmentation pipeline was included to increase class representation and also increase variance in data. After testing out some deep learning models, the "Residual Attention Network for image classification [10] was selected. Its architecture is highly complex, with attention and residual modules stacked in the network. The system uses the intuition of using highlighting the differentiating and unique features in the image for detection. The path splits into a mask and trunk branch. Trunk branch is the attention module that calculates the receptive field in the input for generating attention masks to highlight features. It is highly adaptive and while moving ahead in the network, increases its attention on important features. Mask branches can be viewed as a bottom-up top-down fully convolutional structure. Mask branch aims at improving the features of the trunk branch. The final layer being the classifier is the fully connected neural network with the output channels as per our output classes.

2.3 Feasibility Study

Considering that the project aim was only to augment or complement a medical professional's skills to diagnose diseases, and not replace the doctor entirely, state of the art detection was not the initial target. A satisfactory recall and precision score which would help professionals in remote and rural areas, only taking image data as input was the target. As part of data exploration, it was noted that most data was highly anonymized and much was prohibited for commercial use. Furthermore, much of the data was unreliable and was found to be mislabeled in many cases. Data collection became a key process as wrong data would cause bad results and that could be disastrous in terms of personal loss of the user/patient. The databases used were verified by medical professionals for its labeling, reliability, and usability.

Keeping in mind that the target areas might not have high processing computers or rather no connectivity too, the deployment of such a system would have to support multiple platforms. To support working in low-performance processors the model needs to be efficient and lightweight. Furthermore, an independent module was to be deployed on a specialty processor, in this case, Jetson Nano. Jetson Nano by Nvidia is specifically made AI processor, used to run deep learning models on its efficient processing power. Furthermore, a docker container was also planned to be running on a cloud instance so it can be accessed from any internet connection.

3. SYSTEM ANALYSIS & DESIGN

3.1 Requirement Specification

The **diagnosis system**, in general, should have the following properties:

- Users should be allowed to input only images.
- The system will allow images from a connected camera or from locally saved files.
- The system will show the classification result as a list of 2 or 3 most probable results.
- GUI can be deployed as a docker container or as an independent executable on processors.
- GUI will also output probability as a percentage with every result.

Data Exploration and Preprocessing Pipeline should complete the following requirements:

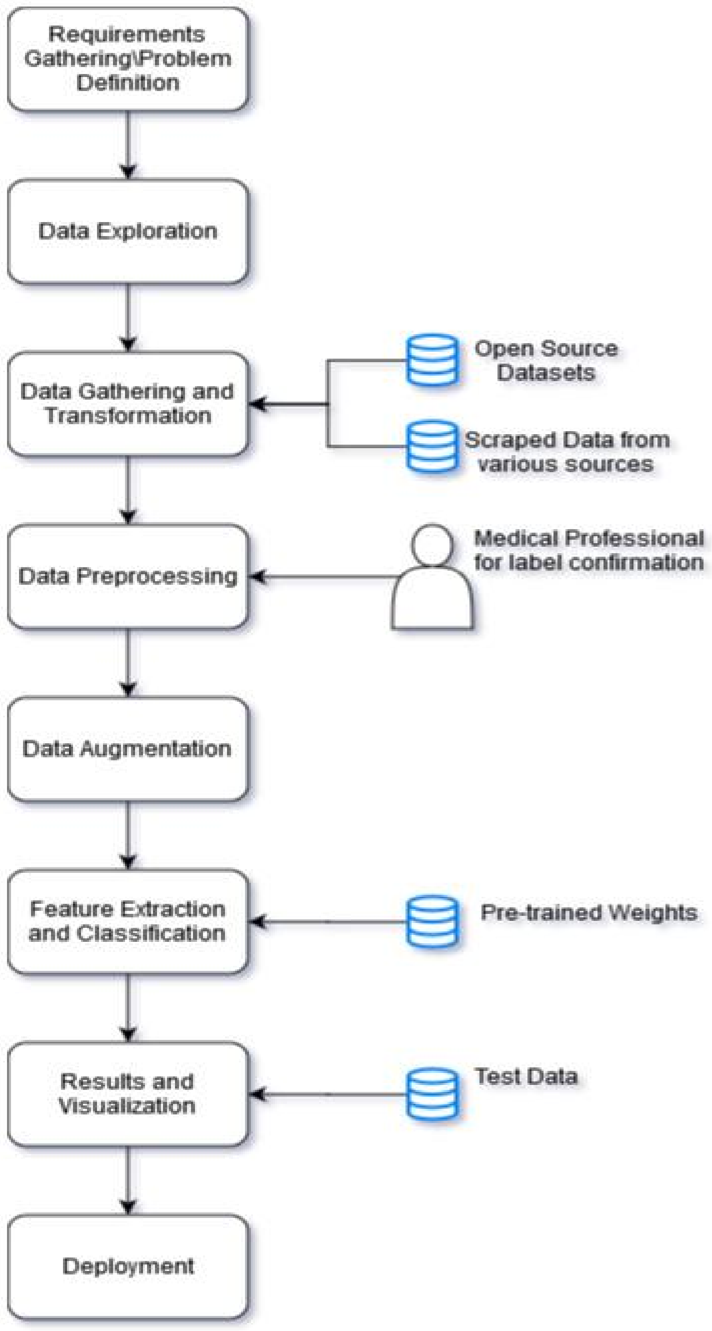
- Labeled images should be verified in a random fashion by medical professionals.
- Images should be open source or licensed appropriately.

Data Augmentation Pipeline should complete the following requirements:

- Training images only to be augmented and separated from test and validation data.
- Data should be licensed for modification.
- Data should be named appropriately for mapping the augmentation techniques used.

3.2 Flowcharts

3.2.1 Development Steps



Requirements Gathering\Problem Definition
Data Exploration
Data Gathering and Transformation
Open Source Datasets
Scraped Data from various sources
Data Preprocessing
Medical Professional for label confirmation
Data Augmentation
Feature Extraction and Classification
Pre-trained Weights
Results and Visualization
Test Data
Deployment

Fig. 1: Flowchart explaining development steps

The above diagram describes the crucial development steps followed similarly for both the modules. Exploratory steps such as studying research and scripting for data transfer are excluded from the diagram. The details of most of these stages are explained further in the report.

3.2.2 Data Augmentation Flow

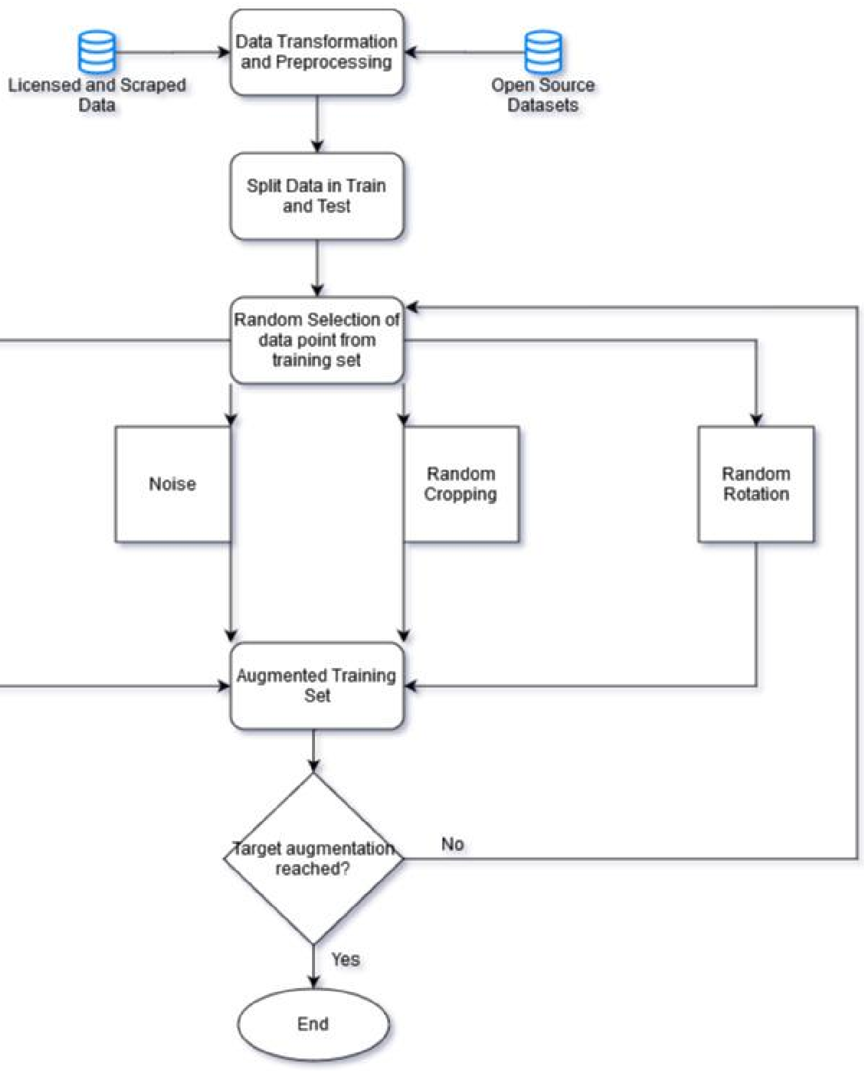

Licensed and Scraped Data
Data Transformation and Preprocessing
Open Source Datasets
Split Data in Train and Test
Random Selection of data point from training set
Image Mirroring
Noise
Random Cropping
Random Rotation
Augmented Training Set
Target augmentation reached?
No
Yes
End

Fig. 2: Flowchart explaining augmentation steps

The above diagram explains the data augmentation process implemented for both the modules. The scripts for the augmentation functions are modified according to the image dataset used and the target augmentation value.

3.3 Design and Test Steps

3.3.1 Requirement Gathering

Skin Disease Diagnosis

As mentioned before, datasets in the medical fraternity are highly anonymized and not for public use to protect the patient's identity. Nevertheless, a public repository, [9] allows the use of its dataset with prior permission. Furthermore, web scraping helped in filtering out open source images for our use. During the gathering of data, it was found that many images were misclassified. Many images were just advertisements. Furthermore, many had low visibility due to low resolution. These issues resulted in extra work in filtering out for useful data. Afterward, the data was randomly selected for testing the reliability(correctly labeled) by a medical professional.

Lung Disease Diagnosis

According to the requirements, only image data was to be used for the classification problem. Usually, x-rays are used to complement the doctor's diagnosis primarily with other tests. To make the system feasible only certain ailments were decided to be classified. The open-source data by NIH Clinical Center [4] was used. The data was highly biased for some classes and the representation of the Pneumonia class was very low. To augment the scarcity of data, another Kaggle dataset [5] was made use of. This data too was randomly verified by the medical professional for its reliability.

3.3.2 Feature Extraction and Classification Phase

Much exploration was conducted for the state of the art research papers in deep learning classifiers in medicine. Many classifiers used segmentation as a former step to further increase accuracy. As our image data only consisted of the diseased part, be it skin images or x-rays, segmentation would not be fruitful. Furthermore, in the skin dataset, as skin diseases are highly variant in the location of the disease, a segmentation task would give improper results and would further worsen the classification results.

As the data to be processed, convolutional neural networks are deemed to be the best for classification. Deep CNNs extract the minute features from an image with ease and furthermore is efficient than manipulating the image data matrix for a classification task with general techniques as Support Vector Machine (SVM) or fully-connected neural networks. Initially a custom small CNN was used to judge the results. Clearly the accuracy was low. Developing the classifier, for each module we followed an iterative process.

Hyperparameter tuning along with experimenting with different loss functions was also followed to achieve better results. As in the skin dataset, the classes were custom and one of a kind, therefore the team could not find a standard benchmark to compare our results with. For the classification part, the models followed already consisted of classifier layers. To increase and adjust to our data, both the modules consisted of changing the classifier layers to our necessities. The VGG model's classifier is trained on ImageNet with 1000 output classes, we changed it according to our number of classes. Furthermore, we made the classifier deeper to achieve higher accuracy. Similarly was the classifier layer changed in the lung disease module. Also, the augmentation target was also prone to experimentation. To make sure the model does not overfit but also making sure we get the best possible results, different class strengths were experimented with.

3.3.3 Presentation and Deployment Phase

As per the user's needs, the prototype for a working deep learning classifier was presented as a simple GUI scripted in Python. The GUI application was developed using Tkinter [11]. As per the requirements, the GUI allowed the user to take an image via the connected camera or use an existing image to view the results. The GUI also allowed a preview of the taken or selected image before the feature extraction and classification. The results are previewed as a probability as a percentage of the top 2-3 most probable results.

Requirements included the need for the prototype to be deployed using various platforms to analyze the feasibility of each. Firstly, the GUI application was developed as part of the Flask [12] API for the local network. Understanding the response time and results, the project was planned for further deployment. According to the requirements, the Flask API was containerized using the Docker platform. The container was hosted on the Amazon EC2 instance and tested successfully. Furthermore, as the users were to be in remote and rural areas, an internet independent platform was planned to deploy as an independent application, which can be transported and accessed easily. Jetson Nano by Nvidia was selected as the appropriate platform for its processing capabilities specially curated for deep learning models. The application was scripted to work with the attached webcam and was tested successfully by the execution of the prototype.

3.4 Algorithms and Pseudo Code

3.4.1 Adam Optimizer Algorithm

Adam optimizer algorithm [13] was used as an optimization technique in both the deep learning modules. The following is the rough algorithm for the optimizer.

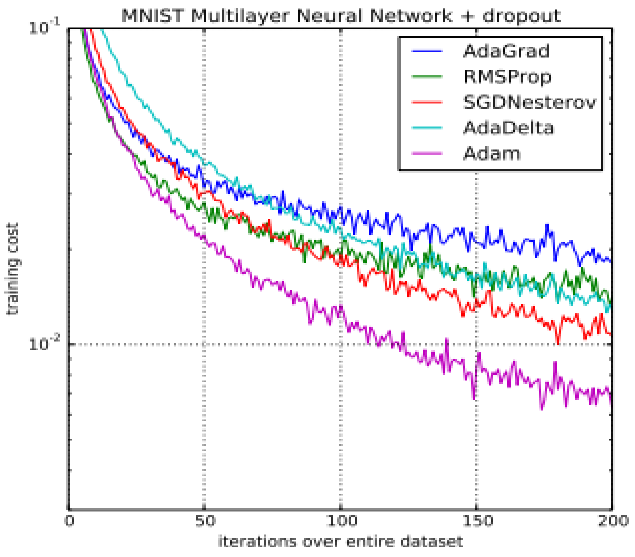


Fig. 3: A comparison between different optimization algorithms

Source: [13]

1. Initialize variables V_{dw}, V_{db}, S_{dw}, S_{db} to zero.
2. Compute dw and db for current batch (Weight and Bias)
3. Update formula for V_{dw}: V_dw = $\beta_1 \times V_{dw} + (1 - \beta_1) \times dW$
4. Update formula for $V_{db} = \beta_1 \times V_{db} + (1 - \beta_1) \times db$
5. Like Rmsprop: $S_{dw} = \beta_2 \times S_{dw} + (1 - \beta_2) \times dW^2$
6. $S_{db} = \beta_2 \times S_{db} + (1 - \beta_2) \times db^2$
7. We also implement bias correction as:
    a. $V_{dw^{corrected}} = V_{dw}/(1 - \beta_1^{\ t})$
    b. $V_{db^{corrected}} = V_{db}/(1 - \beta_1^{\ t})$
    c. $S_{dw^{corrected}} = S_{dw}/(1 - \beta_2^{\ t})$
    d. $S_{db^{corrected}} = S_{db}/(1 - \beta_2^{\ t})$

```
8. Update W and b
    a. $W = W - learning\ rate \times (V_{dw}^{corrected} / \sqrt{S_{dw}^{corrected} + \varepsilon})$
    b. $b = b - learning\ rate \times (V_{db}^{corrected} / \sqrt{S_{db}^{corrected} + \varepsilon})$
```

ε, β_1, β_2 are the hyperparameters. β controls the two exponentially weighted averages.

3.4.2 Cross Entropy Loss

Cross-Entropy Loss was followed as the loss function for the deep learning models. Also known as log loss. It measures the performance of a classification model whose output is in probability.

Pseudocode:

```
1. Input be y (binary indicator; true value), M = number of classes, p = predicted probability,
    a. $Loss = -\sum_{c=1}^{M} y_c log(p_c)$
```

3.5 Testing Process

A crucial part of the development of deep learning models is testing. Accurate and constructive testing was followed with iterative corrections for increased accuracy.

The following testing processes were followed:

1. Ensure sanity of the splitting of train and test sets

The inclusion of data points from the training set to the test set would result in false accuracy measurements as the model will not generalize well with other data points. As the data was already used for training, the model parameters are optimized for the same and will give a positive result when tested with the same data but may perform worse on new unseen data.

2. Following training and validation loss

The model version during the training epochs with the least validation loss is saved as the best model. The validation loss is calculated with the unseen data to make sure the model generalizes well.

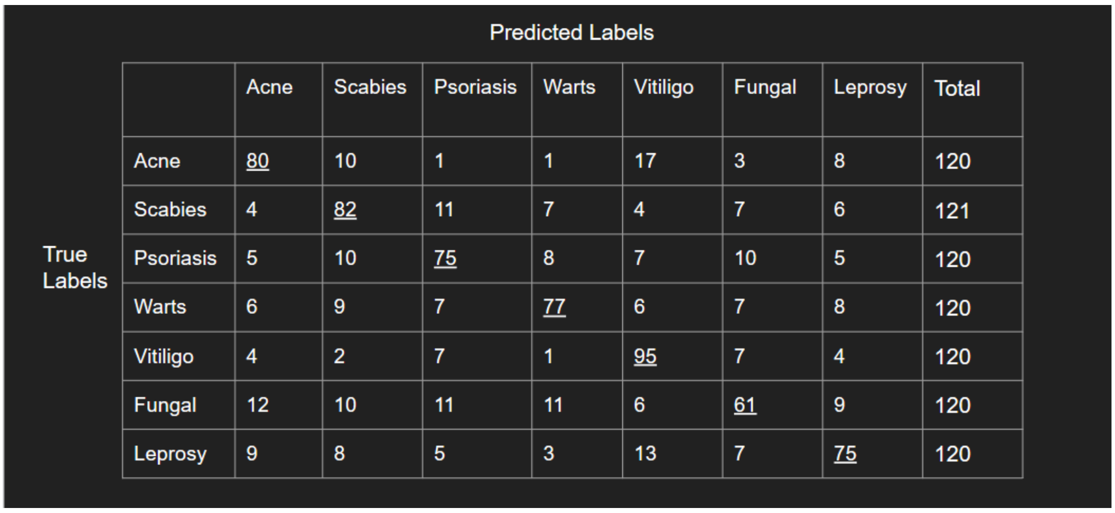
3. Accuracy and Confusion Matrix

The performance metric used is average accuracy. The overall accuracy in a classification model is visualized with the confusion matrix. This matrix helps in finding the low as well as high performing classes.

4. RESULTS & OUTPUTS

4.1 Confusion Matrix

4.1.1 Skin Disease Model

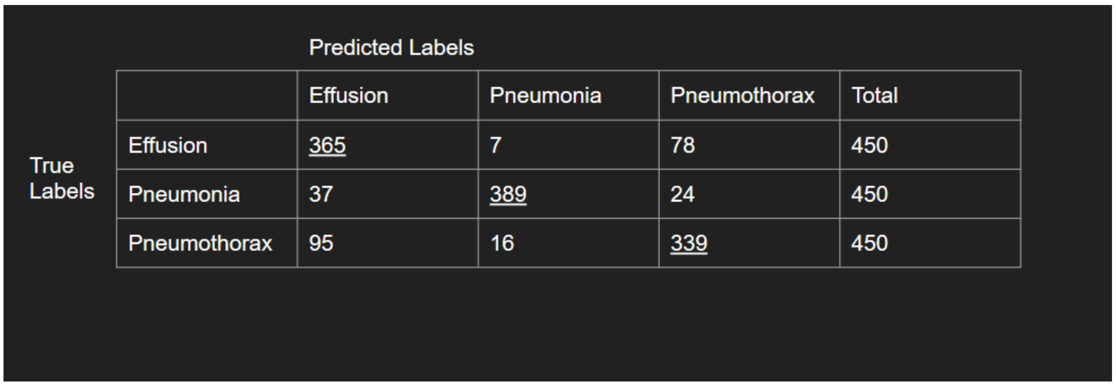


		Predicted Labels							
		Acne	Scabies	Psoriasis	Warts	Vitiligo	Fungal	Leprosy	Total
True Labels	Acne	80	10	1	1	17	3	8	120
	Scabies	4	82	11	7	4	7	6	121
	Psoriasis	5	10	75	8	7	10	5	120
	Warts	6	9	7	77	6	7	8	120
	Vitiligo	4	2	7	1	95	7	4	120
	Fungal	12	10	11	11	6	61	9	120
	Leprosy	9	8	5	3	13	7	75	120

Fig. 4: Confusion Matrix for Skin Diagnosis Model

As seen in the confusion matrix, certain classes perform well while some classes underperform. The overall accuracy is found to be 64.8%. Fungal infections have the highest variance and cannot be localized to one body part. Furthermore, the different infections are recorded inconsistently and therefore have much less standard records. In comparison to the different state of the art benchmarks, this result can be said to be in the vicinity. We discuss in the conclusions, about the future work to improve on these results.

4.1.2 Lung Disease Model



Predicted Labels

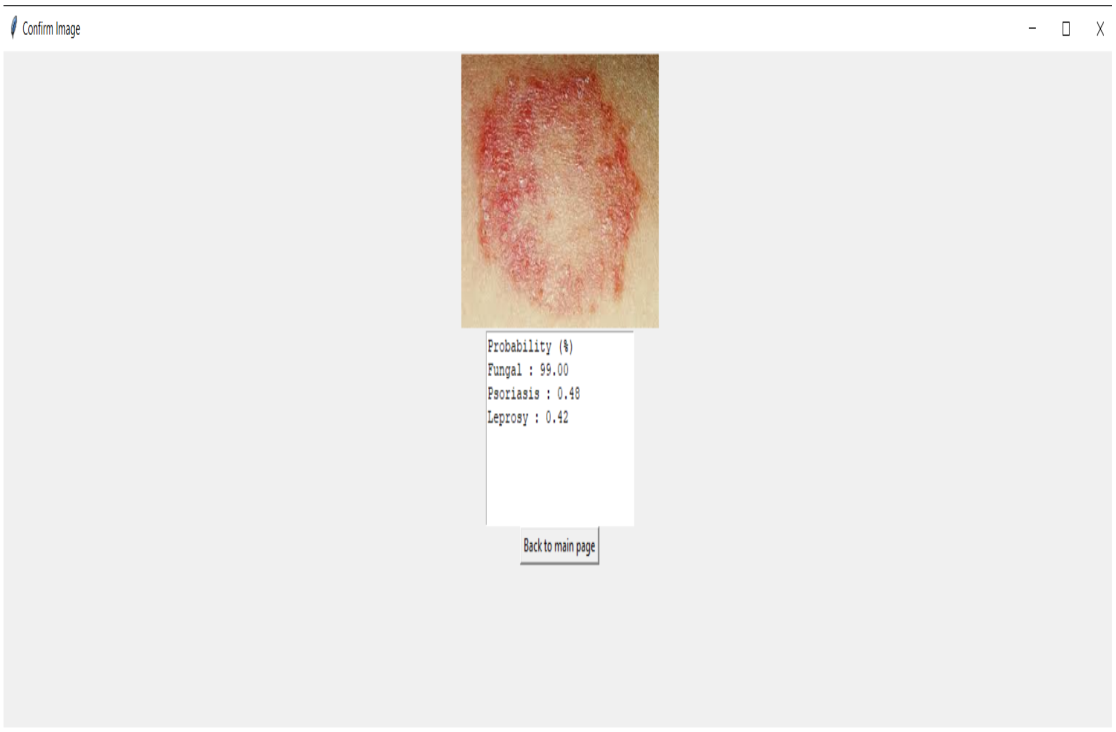
True Labels

	Effusion	Pneumonia	Pneumothorax	Total
Effusion	365	7	78	450
Pneumonia	37	389	24	450
Pneumothorax	95	16	339	450

Fig. 5: Confusion Matrix for Lung Diagnosis Model

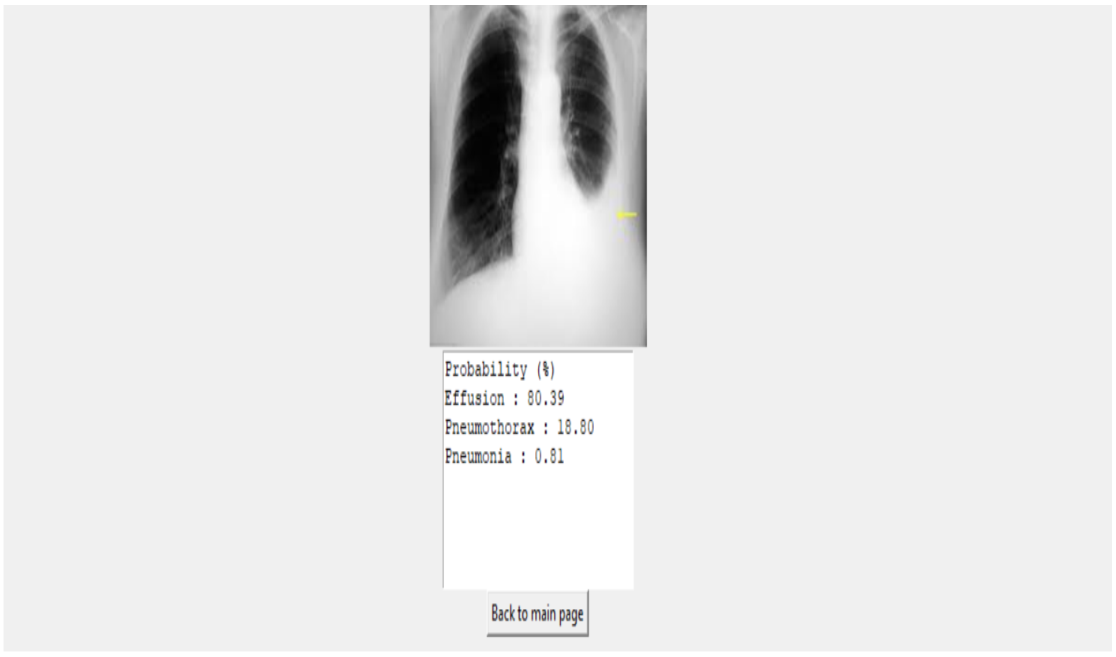
The overall accuracy is found to be around 80%. Considering the state of the art in a similar domain, the results are found to be satisfactory. Furthermore, many systems use more than x-ray information to derive a conclusion. In our case, using only image data, we predict a class.

4.2 Predictive Model GUI



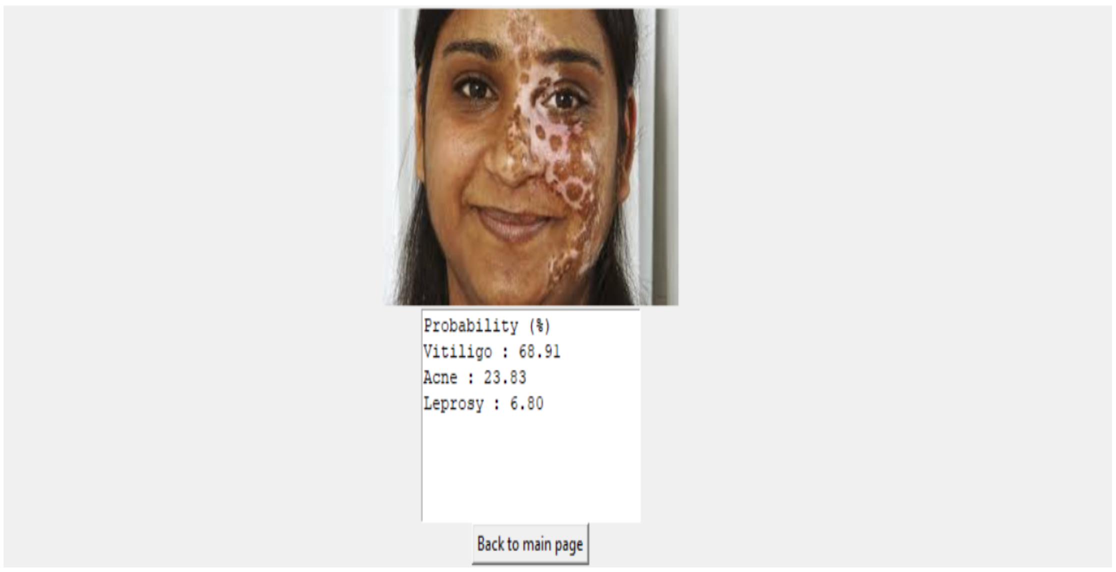
```
Probability (%)
Fungal : 99.00
Psoriasis : 0.48
Leprosy : 0.42
```

Fig. 6: Skin Diagnosis output for Fungal Infection image



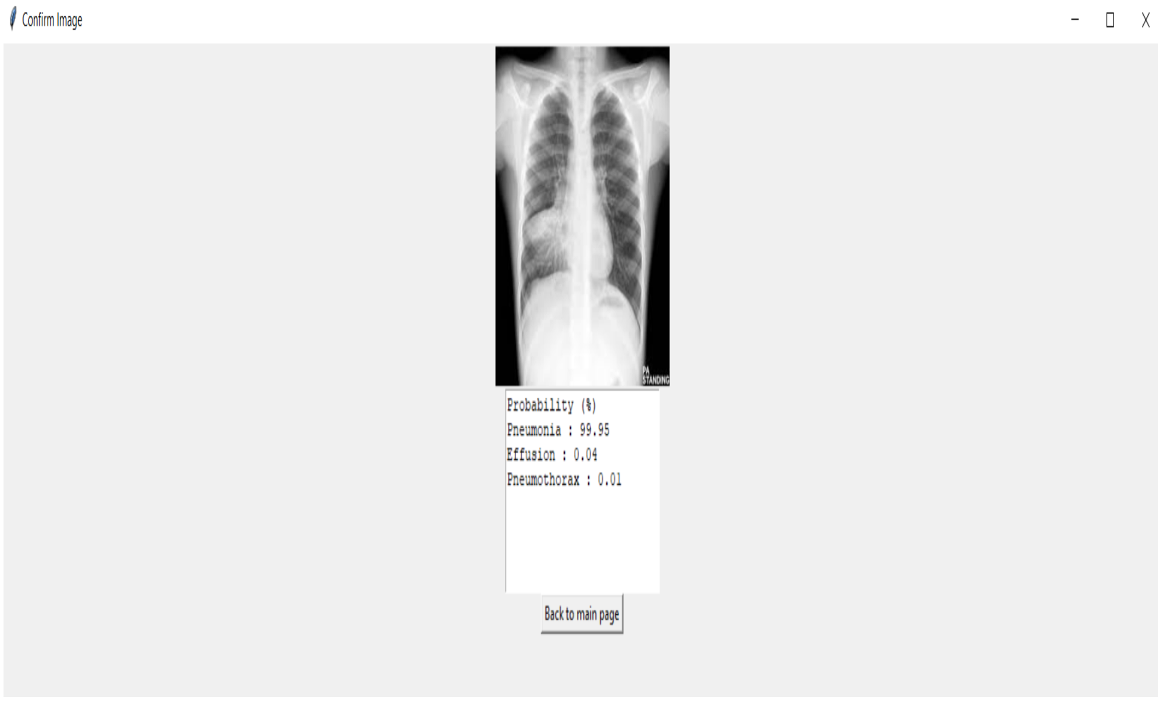
```
Probability (%)
Effusion : 80.39
Pneumothorax : 18.80
Pneumonia : 0.81
```

Fig. 7: Lung Diagnosis output for Effusion image



```
Probability (%)
Vitiligo : 68.91
Acne : 23.83
Leprosy : 6.80
```

Fig. 8: Skin Diagnosis output for vitiligo image

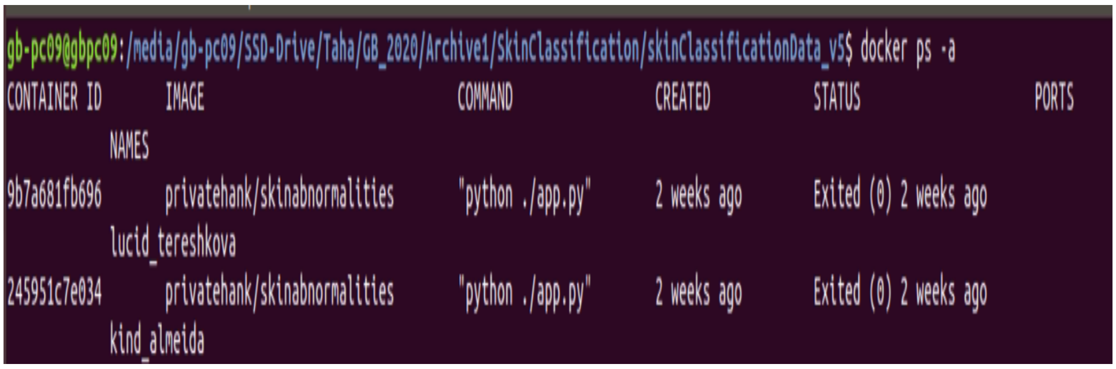


```
Probability (%)
Pneumonia : 99.95
Effusion : 0.04
Pneumothorax : 0.01
```

Fig. 9: Lung Diagnosis output for Pneumonia image

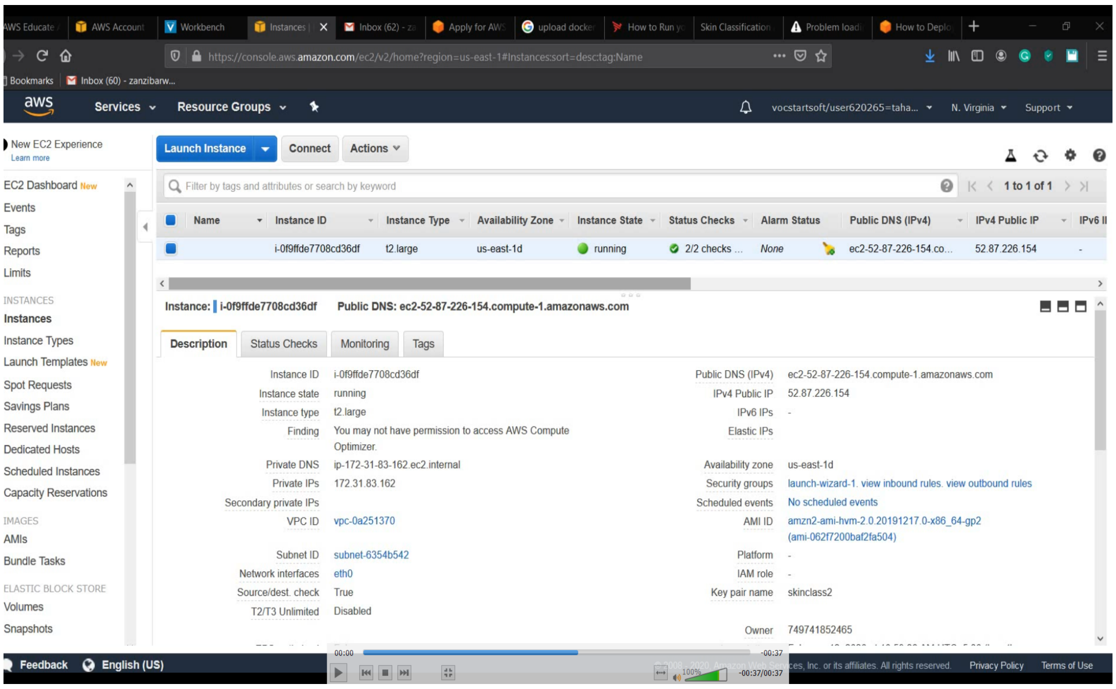
As seen in the above figures, a simple GUI was developed for the prototype. The GUI allowed images captured by the camera or an existing file. The GUI also allows a preview, before the image is selected to be classified. The GUI for lung diagnosis and skin diagnosis was built on different models although it seems to be exactly the same. To this point in the project development, GUI development was not a priority and hence a simple design.

4.3 System Deployment



```
gb-pc09@gbpc09:/media/gb-pc09/SSD-Drive/Taha/GB_2020/Archive1/SkinClassification/skinClassificationData_v5$ docker ps -a
CONTAINER ID        IMAGE                           COMMAND             CREATED             STATUS                   PORTS
          NAMES
9b7a681fb696        privatehank/skinabnormalities   "python ./app.py"   2 weeks ago         Exited (0) 2 weeks ago
          lucid_tereshkova
245951c7e034        privatehank/skinabnormalities   "python ./app.py"   2 weeks ago         Exited (0) 2 weeks ago
          kind_almeida
```

Fig. 10: Output for 'docker ps -a'



https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#Instances:sort=desc:tag:Name

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6 I
	i-0f9ffde7708cd36df	t2.large	us-east-1d	running	2/2 checks ...	None	ec2-52-87-226-154.co...	52.87.226.154	-

Instance: i-0f9ffde7708cd36df   Public DNS: ec2-52-87-226-154.compute-1.amazonaws.com

Instance ID	i-0f9ffde7708cd36df	Public DNS (IPv4)	ec2-52-87-226-154.compute-1.amazonaws.com
Instance state	running	IPv4 Public IP	52.87.226.154
Instance type	t2.large	IPv6 IPs	-
Finding	You may not have permission to access AWS Compute Optimizer.	Elastic IPs	
Private DNS	ip-172-31-83-162.ec2.internal	Availability zone	us-east-1d
Private IPs	172.31.83.162	Security groups	launch-wizard-1. view inbound rules. view outbound rules
Secondary private IPs		Scheduled events	No scheduled events
VPC ID	vpc-0a251370	AMI ID	amzn2-ami-hvm-2.0.20191217.0-x86_64-gp2 (ami-062f7200baf2fa504)
Subnet ID	subnet-6354b542	Platform	-
Network interfaces	eth0	IAM role	-
Source/dest. check	True	Key pair name	skinclass2
T2/T3 Unlimited	Disabled	Owner	749741852465

Fig. 11: Amazon EC2 instance dashboard

5. CONCLUSION

To help with a faster and reliable diagnosis for rural and remote areas, to work for a good cause was one of the main motivations to keep giving effort for the project. Understanding the vulnerability of such a diagnosis system and also with the risks involved from a user perspective, the foremost aim of the team was to produce a product which only augmented or complemented the skills of a medical professional.

The skin diagnosis model, although promising, still needs to improve for satisfactory results. One of the major issues was the lack of standard and reliable data to train our computer vision models. Also, as the classes were decided after some discussions regarding the Indian medical environment and the spread of different diseases, this specific system was one of a kind. In the lung diagnosis model, the accuracy was satisfactory, nevertheless the team aims to improve it further.

The project spanned all domains of deep learning development. From data gathering to deployment, every stage was well defined. Furthermore, enough room was given to me and the team to grow, but I also make mistakes and learn from them. Many stages or concepts were new altogether and enough exploratory resources were made use of to learn and implement. In these trying times of the ongoing pandemic, the importance of diagnostic science is realized and such projects, although a small step, give promise of the positive influence artificial intelligence and deep learning will have on medical science.

6. REFERENCES

[1]. Tschandl, Philipp, 2018, "The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions", https://doi.org/10.7910/DVN/DBW86T, Harvard Dataverse, V1, UNF:6:IQTf5Cb+3EzwZ95U5r0hnQ== [fileUNF]

[2]. Howard, Andrew G., et al. "Mobilenets: Efficient convolutional neural networks for mobile vision applications." *arXiv preprint arXiv:1704.04861* (2017).

[3]. Wei, Li-sheng, Quan Gan, and Tao Ji. "Skin disease recognition method based on image color and texture features." *Computational and mathematical methods in medicine* 2018 (2018).

[4]. Wang X, Peng Y, Lu L, Lu Z, Bagheri M, Summers RM. ChestX-ray8: Hospital-scale Chest X-ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases. IEEE CVPR 2017,http://openaccess.thecvf.com/content_cvpr_2017/papers/Wang_ChestX-ray8_Hospital-Scale_Chest_CVPR_2017_paper.pdf

[5]. https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia

[6]. Rajpurkar, Pranav, et al. "Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning." *arXiv preprint arXiv:1711.05225* (2017).

[7]. Huang, Gao, et al. "Densely connected convolutional networks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.

[8]. Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556* (2014).

[9]. DermNet, N. Z. "DermNet NZ website." (2008).

[10]. Wang, Fei, et al. "Residual attention network for image classification." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017.

[11]. "Tkinter - Python Interface to Tcl/Tk." Tkinter - Python Interface to Tcl/Tk - Python 3.8.2 Documentation, docs.python.org/3/library/tkinter.html.

[12]. "Welcome to Flask." Welcome to Flask - Flask Documentation (1.1.x), flask.palletsprojects.com/en/1.1.x/.

[13]. Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." *arXiv preprint arXiv:1412.6980* (2014).

7. APPENDICES

7.1 Image Classification using Deep Learning

The task of image classification or image recognition in the industry in the current scenario is dominated by deep convolutional neural networks. The primary CNNs for image classification encountered are:

- VGGNet [8]
- DenseNet [7]
- MobileNet [2]

VGG Neural Networks were the early CNN architectures, which took care of the 'depth' of the network. Higher the depth, the more the trainable parameters. With the right amount of data, one could easily take advantage of these trainable parameters to solve a complex image classification task. The classification layer at the end of the network could easily be replaced with many readily available libraries. The model achieves a 92.7% top-5 test accuracy on ImageNet.

In DenseNet, each layer obtains inputs from all the previous layers. Furthermore, each layer passes its feature-map to its consecutive layers. This process is called concatenation. As previous layers are used, the model is thinner as the number of trainable parameters and channels are fewer. This architecture allows the error flow to be propagated to higher levels and thus the weight adjusts much faster and more accurately. It achieved the state of the art results on ImageNet test data. This model is computational heavy as the parameters increase every layer. Furthermore, the memory used is higher too as we have to store feature maps from every output to be passed and also during backpropagation.

MobileNets used the concept of depthwise separable convolutions to increase the model's computational efficiency. The method includes depthwise convolution followed by pointwise convolution. The method dramatically decreases the trainable parameters in the convolutional stage by separating the convolution part in two stages. Depthwise convolution is channel-wise spatial convolution and pointwise convolution is 1x1 convolution to change the shape of the output. MobileNet is known for its lightweight models and was a top consideration in the project to be implemented on the independent platform. Less complexity meant that complex classification tasks of the skin dataset would not be learned for optimum results.