University of the Philippines Manila
College of Arts and Sciences
Department of Physical Sciences and Mathematics

HeartSmart: A Clinical Decision Support System for Cardiomegaly Detection in Chest X-ray Images using Convolutional Neural Network

A special problem in partial fulfillment
of the requirements for the degree of
Bachelor of Science in Computer Science

Submitted by:

Edward Francis L. Lacanlale
June 2018

Permission is given for the following people to have access to this SP:

Available to the general public	Yes
Available only after consultation with author/SP adviser	No
Available only to those bound by confidentiality agreement	No

ACCEPTANCE SHEET

The Special Problem entitled "HeartSmart: A Clinical Decision Support System for Cardiomegaly Detection in Chest X-ray Images using Convolutional Neural Network" prepared and submitted by Edward Francis L. Lacanlale in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science has been examined and is recommended for acceptance.

Marvin John C. Ignacio, M.S. (*cand.*)
Adviser

EXAMINERS:

	Approved	**Disapproved**
1. Gregorio B. Baes, Ph.D. (*cand.*)	_____	_____
2. Avegail D. Carpio, M.Sc.	_____	_____
3. Richard Bryann L. Chua, Ph.D. (*cand.*)	_____	_____
4. Perlita E. Gasmen, M.Sc. (*cand.*)	_____	_____
5. Ma. Sheila A. Magboo, M.S.	_____	_____
6. Vincent Peter C. Magboo, M.D., M.Sc.	_____	_____
7. Geoffrey A. Solano, Ph.D (*cand.*)	_____	_____

Accepted and approved as partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science.

Ma. Sheila A. Magboo, M.Sc.
Unit Head
Mathematical and Computing Sciences Unit
Department of Physical Sciences and Mathematics

Marcelina B. Lirazan, Ph.D.
Chair
Department of Physical Sciences and Mathematics

Leonardo R. Estacio Jr., Ph.D.
Dean
College of Arts and Sciences

Abstract

Cardiovascular disease is the Philippines' top cause of mortality. CVD typically results to high blood pressure and enlargement of the heart or cardiomegaly. HeartSmart is a web-based decision support tool. It assesses the patient's risk of heart enlargement with chest x-ray image as an input through the use of a convolutional neural network. The trained model in the system gives an accuracy of 71%, which can be further improved before it can be used as a decision support tool for detecting cardiomegaly in patients.

Keywords: cardiomegaly, convolutional neural network, decision support system

Contents

Acceptance Sheet **i**

Abstract **ii**

I. Introduction **1**

A. Background of the Study 1

B. Statement of the Problem 9

C. Objectives of the Study 9

D. Significance of the Project 10

E. Scope and Limitations 11

F. Assumptions 11

II. Review of Related Literature **12**

III. Theoretical Framework **16**

A. Cardiomegaly 16

B. Chest X-Ray 17

C. Clinical Decision Support System 19

D. Telemedicine 19

E. Convolutional Neural Network 19

IV. Design and Implementation **24**

A. System Overview 24

B. Use Case Diagram 25

C. Entity Relationship Diagram 26

D. Activity Diagram 27

1. Rural Health Worker 27

2. Radiologist 28

3. AI Expert 29
E. Technical Architecture 30
F. Dataset 30

V. Results **31**
A. Home Screen 31
B. Standard User's Account 32
C. Doctor's Account 35
D. AI Expert's Account 36

VI. Discussions **41**

VII. Conclusions **43**

VIII. Recommendations **44**

IX. Bibliography **45**

X. Appendix **51**
A. Source Code 51

XI. Acknowledgement **63**

I. Introduction

A. Background of the Study

Cardiovascular disease is the Philippines' top cause of mortality. Out of 514,745 deaths registered in 2013, it recorded a 22.3% share from total deaths [1] and the numbers continue to grow within the succeeding years. Cardiovascular disease (CVD) is a type of non-communicable disease that involves the heart or blood vessels [2]. In our country, the most prevalent type of CVD is the coronary heart disease [3]. It is normally caused by accumulation of cholesterol in the arterial wall, narrowing the arteries, and reducing the blood flow to the heart [4]. This condition typically results to high blood pressure and enlargement of the heart.

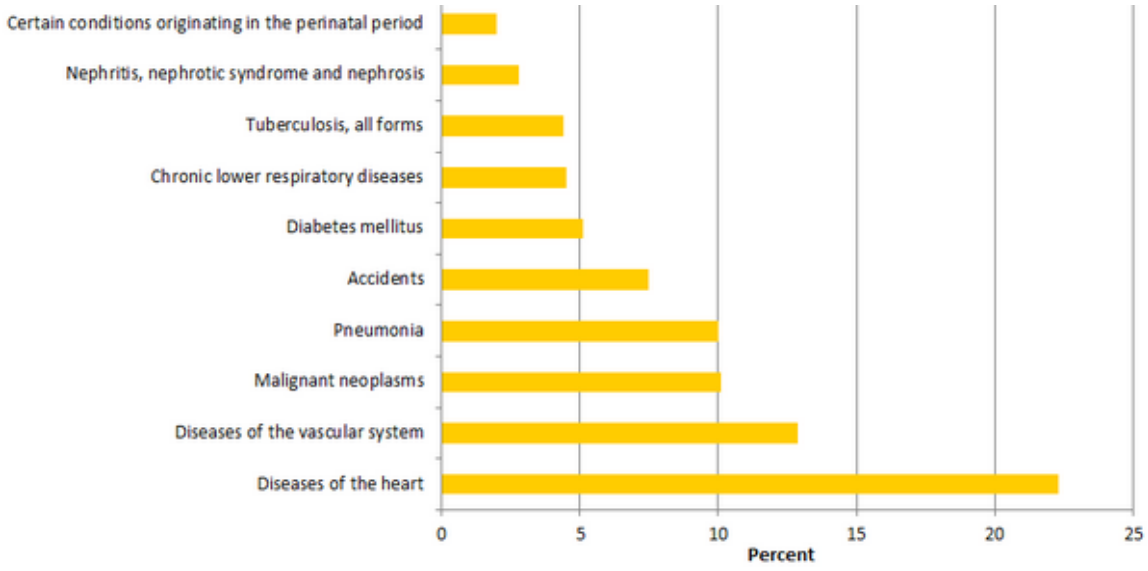


Figure 1: Top causes of mortality, 2013

An enlarged heart may be discovered due to recurring symptoms tied with coronary heart disease which may later develop into severe heart failure.[5] Heart Failure (HF) can be defined as an abnormality of the cardiac structure or function leading to failure of the heart to deliver required amount of blood and oxygen throughout the body. [6] As heart tries to make up for difficulty of pumping enough blood, the heart

stretches to contract strongly and keep up with the demand to pump more blood which results to its enlargement. [7]

Heart enlargement due to HF can be diagnosed if suspicions are present. Heart enlargement can be detected via different means. Electrocardiogram, echocardiogram, stress test, CT or MRI scan to name a few. But since chest x-ray is typically used when patient shows onset symptoms of HF, [7] it is commonly used as the preliminary indicator if the patient has an enlarged heart.[8]

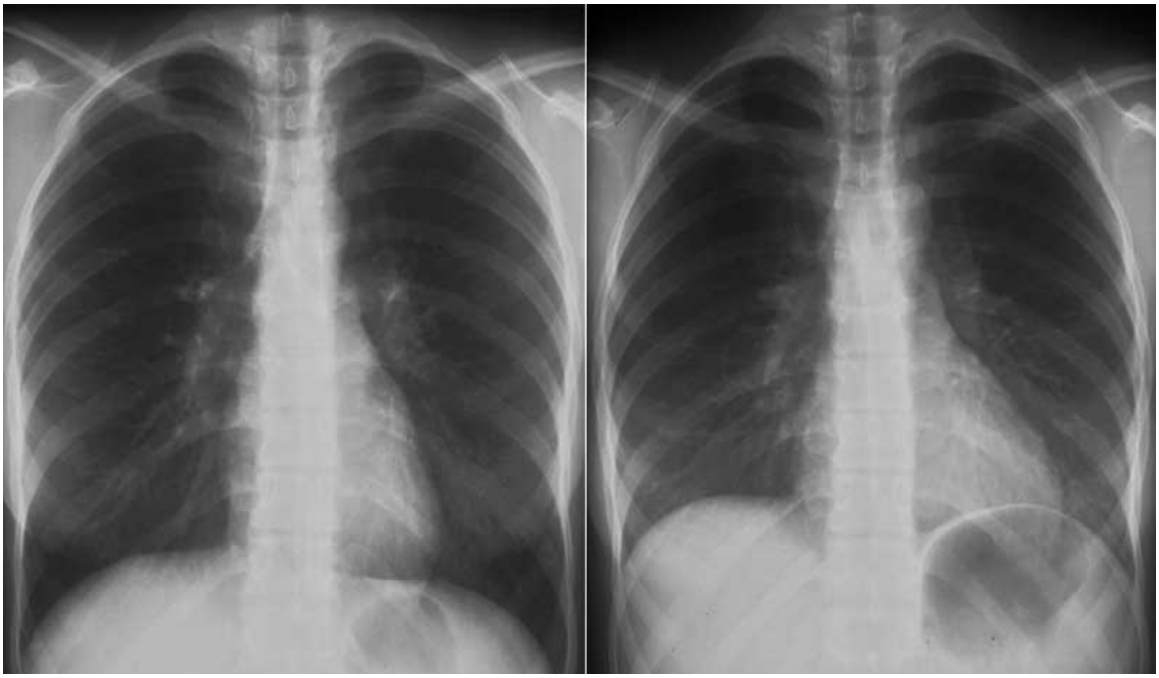


Figure 2: Chest x-ray image with normal (*left*) and enlarged (*right*) heart

Chest x-ray (CXR) is a fast and painless imaging test that uses x-ray waves to create pictures of the structures in and around your chest. [9] Common CXR diagnosis report involves the condition of lungs, heart, and mediastinum. If an abnormality is present in one of those findings, other tests are also conducted to verify the result.

In all branches of medicine, there is always an inevitable element of patient exposure to problems arising from human error [10]. Discrepancies in medical diagnosis does not only tarnishes the reputation of the medical institution involved, but more

importantly, it places the patients life at risk. Some of the factors that causes these discrepancies are volume of the workload, complexity of the image being analyzed, and negligence to the small details.

A research conducted by Monfared et al shows that manually diagnosing cardiomegaly through CXR had an estimated sensitivity rate of 34% and specificity rate of 84.5%. The study concluded that all patients with suspected heart failure should undergo echocardiography. Although CXR may not have the same diagnostic accuracy as echocardiography, its easy accessibility and high specificity in diagnosis of cardiomegaly is very helpful, particularly in screening the enlarged heart size. [11]

Although many errors are little or no significance to the patient, it is advisable to eradicate any room for error in medical diagnosis. In order to prevent the aforementioned inaccuracies, it is suggested to always have a highly trained radiologist present, double check the result done by the attending radiologist, or use computer-aided diagnosis.[10]

In quest to build a more efficient tool for medical diagnosis, most medical institution and research laboratories use artificial intelligence theories (AI) to real-world applications.[12] Traditional AI is usually created by giving a set of rules and applying it to the targeted problem to be solved [13]. But in order to keep up with the demands of revolutionizing healthcare, adaptive technology is needed.

Machine learning is the capability of an AI system to improve its performance over a period of time. It assumes the capability of the system to acquire new knowledge and skills. Integrating this to the current demand of technological advancement of medicine, ML as a diagnostic tool will generate incredible efficiency, generates less cost, and most importantly, if properly implemented, can save lives. [14] [12]

However, there is a downside with this approach. The performance and accuracy of ML methods rely on the features of the data as its input.[15] Feature is defined as a piece of information used as a representation of the data. In an image, a feature

may be represented as points, edges, or objects. Feature extraction, preprocessing of input data, and actual image classification use seperate algorithms. This is process is labor-intensive and time-expensive.[15]

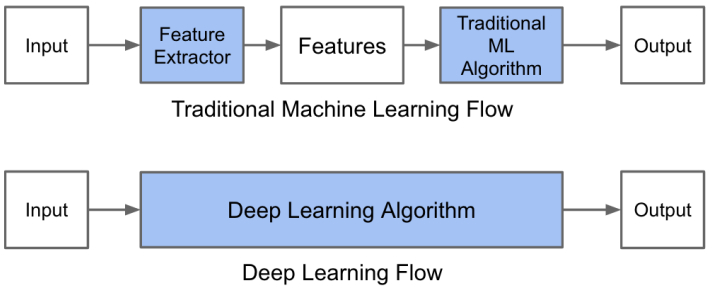


Figure 3: Traditional Machine Learning Flow vs Deep Learning Flow

Deep learning (DL) algorithms, on the other hand, is a class of ML techniques that try to learn high-level features from data. DL allows the computer to build complex concepts out of simpler concepts. [16] This is the distinction and advantage of DL over traditional ML methods. Fig 4 shows how a DL model represents and breaks down complex features into simpler parts.

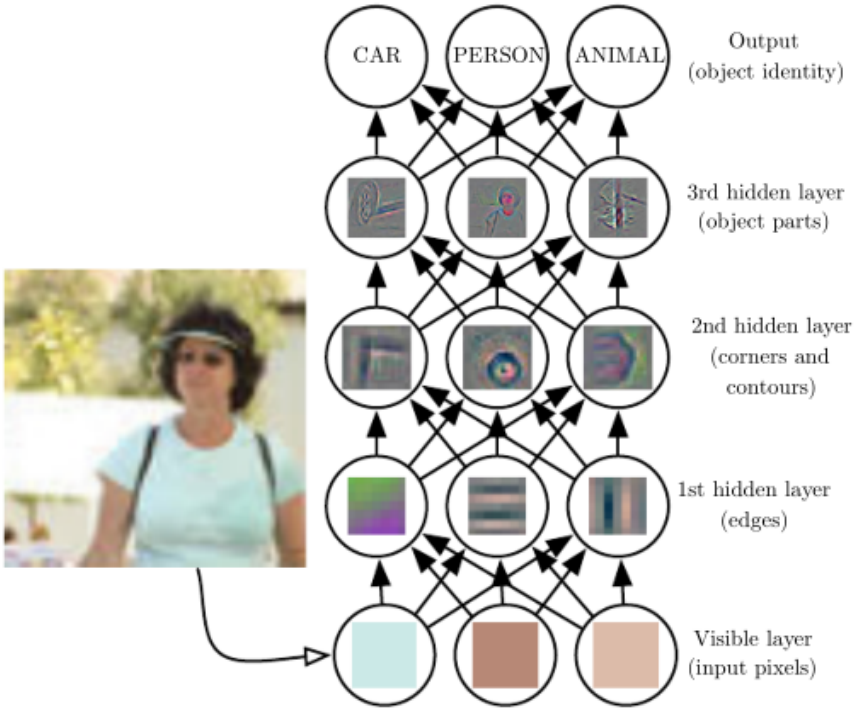


Figure 4: Illustration of a Deep Learning Model

One architecture of DL that is well-suited for object recognition and image classification is the Convolutional Neural Network (CNN). [17] CNN is a special case of Artificial Neural Network that uses convolution operation.

In training the network, the samples are passed through the network and the output obtained is compared with the actual value. The error or the difference in the actual and obtained result is used to change the weights of the link of each neuron. The goal of this learning task is to reach the weight of each link wherein the output provided has minimal error. In order to train the network better and faster, parameter tuning is conducted.These parameters are called hyperparameters.

The common hyperparameters are epoch, learning rate, and batch size. Epoch describes how many times the training data will be procesed by the network. Learning rate defines the speed of convergence of weight to the optimal value. Batch size defines the amount of input vector to be trained simultaneously in the network. This utilizes

the hardware and resources efficiently.

This advantage of DL over traditional ML method has been proven in an annual competition that aims to improve and develop algorithms that could mimic the human vision system. [18]

ImageNet Large Scale Visual Recognition Challenge (ILSVRC) is a contest that has been run annually from 2010. This is considered as the benchmark in object category classification and detection. The goal is to develop an algorithm that achieves high accuracy on several visual recognition tasks using the large-scale image database designed for research provided by the ImageNet project. [19]

From 2010 to 2011, traditional ML methods was used, achieving 28% - 26% error rate in visual recognition tasks. In 2012, a variation of CNN architecture was developed and produced a significantly lower error rate of 16%. With this, researchers became more inclined to use CNN as it produces lower error rates in the succeeding years of the competetion. In the latest ILSVRC result, the winning team achieved an error rate of 2.3%. [20] This shows that CNNs outperform traditional ML methods and gives the best performance in pattern/image recognition problems.

CNN is also a prominent architecture to be used in developing predictive models. Most of these models are integrated into decision support system that is beneficial in the medical field. [21]

Decision Support System (DSS) is defined as an interactive, flexible, and adaptable computer-based information system. This is commonly developed for the purpose of providing solution to management problems.[22] It utilizes known data, usually built with easy-to-use interface, and allows for decision-maker's insight. In the field of healthcare, it is commonly implemented as software that helps in clinical decision-making, often used in matching the characteristics of an individual patient to a clinical knowledge base and patient-specific assessment are then presented as an output for clinical decision. [21] The main advantage of this system is adding efficiency of the

healthcare providers.

In an effort to provide a more accessible health care for Filipinos, Department of Health (DOH) is now using technology to improve the quality of healthcare provided.[23] Supporting the initial *Doctor-to-the-Barrio* program of DOH, telemedicine was introduced and now under continuous development of University of the Philippines Manila - National Telehealth Center (UPM-NTHC).

The UPM-NTHC is the leading research unit in the University of the Philippines responsible for developing cost effective tools and innovations in the realm of information and communications technology (ICT) for improving health care.[24] One of the major breakthroughs of NTHC is the emergence of telemedicine research in remote and underserved areas from Batanes to Zamboanga through a collaborative effort with different government and non-government institutions. This platform has the "potential to reduce the effort and improve the access to health care in rural and other medically unserved and underserved areas."

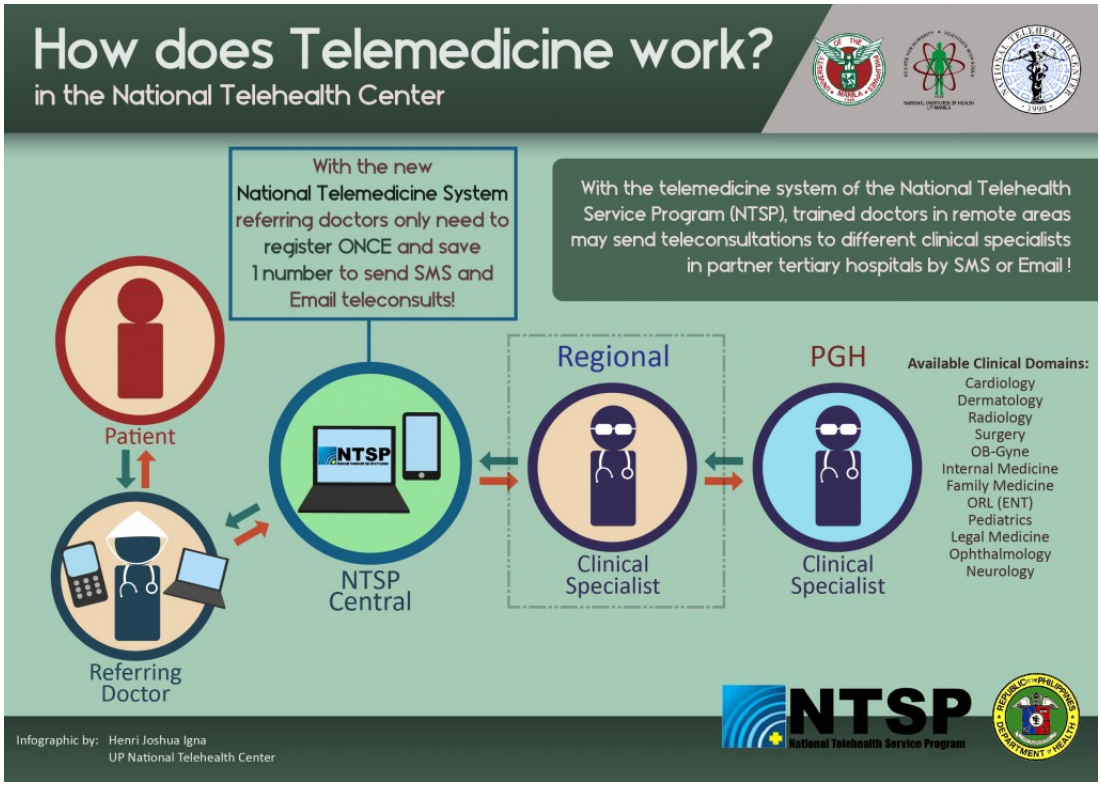


Figure 5: Information Exchange in a Telemedicine

Telemedicine is the use of medical information exchanged from one site to another via electronic communications to improve a patients clinical health status. [25] This includes formulating diagnosis and decision on treatment based on the data and health information transmitted via telecommunications system (eg. SMS, eMail, smart phones, video conferencing, etc.). Fig 5 illustrates the exchange of information in a typical teleconsultation from the referring physician at the rural health unit to the clinical specialist in a partner tertiary health facility (eg. Philippine General Hospital). [26]

The need in telemedicine was due to lack of doctors and specialists in rural and remote areas in the Philippines. In 2015, the estimated total number of radiologists in the Philippines is 1,500 that could provide help to 100 million Filipinos. [23] With the emergence of technology, doctors could provide results even in far-flung hospitals.

B. Statement of the Problem

A research done by Monfared et al, result shows that manual diagnosis of cardiomegaly in CXR yields a low sensitivity rate. In order to lessen the rate of misdiagnosis, radiologists carefully examine the image to provide more accurate diagnosis that often delays result acquisition. This could be worse in remote areas and rural health centers where lack of specialist is a common problem. Mobile x-ray machines exist in rural communities with radiologic technologists operating the machine but diagnosis is still performed by a specialist which commonly reside in an urban community.

C. Objectives of the Study

This study aims to create a web-based system to be used by radiologist to serve as a decision-support system in diagnosing the patient's risk in heart enlargement, to be used by rural health worker for teleconsultations, and to be used by radiology students to serve as a reference in assessing heart condition in chest x-rays. The system has the following functionalities:

1. Allow the health worker or student to:
    (a) Perform image classification (normal or enlarged)
        i. Open PNG image as input to the system
        ii. See the classification result (normal or enlarged) generated by the system
    (b) Consult to a radiologist
        i. Send a message after the result be presented
        ii. Receive a reply regarding the sent message to the doctor
2. Allow the radiologist to:

(a) Perform image classification (normal or enlarged)

i. Open PNG image as input to the system

ii. See the classification result (normal or enlarged) generated by the system

(b) Review and correct the evaluation done by the system

i. Provide annotation once the system provide an incorrect assessment.

(c) Respond to students' or rural health workers' messages

3. Allow the AI expert to:

(a) Prepare or update the dataset

i. Create training and validation set partition

ii. Transfer the images to Caffe's data folder

iii. Convert the dataset into LMDB and compute image mean

(b) Create or retrain the CNN model

i. Select the Neural Network (NN) Architecture to use

ii. Select the prototxts to be used

iii. Begin Feed Forward and Back Propagation for training the NN

iv. Display the training error and validation error for each epoch

v. Save the trained NN as a model

(c) Evaluate the created model

i. Perform batch prediction using the test dataset

D. Significance of the Project

A solution to the need of a more accessible healthcare in rural areas can be attained through the utilization of modern technology. This system will allow the

rural health workers to focus to cases with higher risk. This will also allow radiologists to obtain information to help them correct potential misdiagnosis of the system. Lastly, helping radiology students in assessing patient's heart condition through x-ray images. The system will serve as a decision-making support tool that will allow the rural health worker to have an immediate information in assessing its patients risk level, allowing them to perform other diagnostic tests if needed.

E. Scope and Limitations

1. The input is limited to PNG files of standard CXR film of any age.
2. Other abnormalities of the heart and CXR are not included.
3. The system will provide the probability of normality/abnormality of the image.
4. Those with other anatomic invariants (such as dextrocardia) are not included in the system.
5. The backend used in providing predictive model is Caffe Framework.
6. AI Experts are preregistered in the system.

F. Assumptions

1. The system will only serve as a decision support system, an aid in diagnostic. The final diagnosis will be based on the radiologist from UPM-NTHC.
2. The AI Expert is knowledgeable in creating the prototxts to be used.
3. The AI Expert is knowledgeable in Caffe Framework

II. Review of Related Literature

Heart enlargement is a direct sign of cardiovascular disease. This is the main reason why a lot of researchers are interested in creating a system capable of automated detection of heart enlargement. Most studies use chest x-ray as it is the most common and most accessible diagnostic examination [8]. A study by Carrillo-de-Gea et al proposed a computer-aided detection (CAD) system for CXR images. They presented a novel approach to detect normality or pathology in CXR images. Region of Interest (ROI) was manually labeled using template matching and local binary pattern (LBP) are computed along those area. LBP histogram in a classifier algorithm was used in to attain normality/pathology decision. The approach presented yielded success rates above 87% in the best cases [27]. This shows that a CAD system can reliably assists radiologist in decision-making.

Recently, a lot of paper that uses medical imaging apply convolutional neural network (CNN). This method is commonly used to utilize its ability to automatically identify and classify features and patterns in an image. This is presented in a study by Bar et al. The study used deep learning approach to detect chest pathology in CXR images[28]. Using a CNN model pre-trained with non-medical images, the algorithm is tested on a 433 image dataset to identify different types of pathologies present. Obtaining an area under the curve (AUC) of 0.87-0.94, this result demonstrates the feasibility of detecting chest x-ray pathologies using deep learning approach.

A research by Tataru et al used deep learning for abnormality detection in CXR images [29]. The study compared the performance and accuracy of different CNN architectures using the same dataset. The training data used was 50,000 labeled CXR images with 65:35 normal-abnormal ratio. GoogLeNet network architecture achieves significantly above-random accuracy when distinguishing between normal and abnormal CXR images among the other architectures used. The network is able to attain an accuracy of 0.8.

Recently, Rajpurkar et al developed an algorithm that can detect 14 different diseases from chest x-ray, including cardiomegaly. The main focus of the study is to detect pneumonia at a level exceeding practicing cardiologist. [30] The algorithm, CheXNet, consists of 121-layer CNN trained on a dataset containing over 100,000 frontal x-ray images annotated with 14 diseases. Four practicing academic radiologists annotate a test set, on which the researchers compare the performance of CheXNet to that of radiologists. The result shows that CheXNet exceeds average radiologists performance on pneumonia detection on both sensitivity and specificity. The accuracy on diagnosing the existence of other diseases range from 72.04% to 93.87%

In medical imaging, it is desirable to distinguish features that contributed most to certain abnormality present in the image. Different approaches are used to locate the ROI for classification. One approach is image localization.To localize an image is to occlude a patch in the image to measure its impact on classification. [31] In a simpler notation, it is used to create a "bounding box" surrounding the ROI.

One study done by Islam et al uses localization of feature responsible for classification decision in abnormalities present in CXR [31]. The researchers found out that the network used in the study can successfully localize the abnormalities most of the time, especially for spatially spread out abnormalities like cardiomegaly and pulmunary edema. They also found out that there are characteristic features in the shape of the heart and its surrounding regions that alone is sufficient to detect cardiomegaly. This is comparing to the usual method of using the ratio of heart and lung area as a measure for cardiomegaly.

Another way in obtaining the ROI is through segmentation. Segmentation is achieved by removing a part of an image to the original. This is a crucial step to acquire an effective computer-aided detection on medical images.

A study done by Dai et al propose a framework for organ segmentation in CXR. Using only very limited training data available, the framework reaches human-level

performance in segmenting the ROI without relying on any existing trained model or dataset [32].

A similar research by Avendi et al employed a combination of CNN and deformable models to develop a segmentation tool of left ventricle (LV) in cardiac magnetic resonance imaging (MRI) dataset [33].

A study of Zhao et al used CNN in segmentation and detection of brain tumors [34]. They designed a three-pathway framework which is adaptive to accurate segmentation and classification of brain tumor in MRI images.

Remote healthcare systems have received increasing attention within the last decades [35], with concern in telemedicine as whether the same level of care can be provided without the physical interaction that meeting with a healthcare practitioner in person provides.

A research by Abo-Zahhad et al provided a telemedicine system that includes continuous collection and evaluation of multiple vital signs. The study aims for a long-tem health care centered in remote areas where health care is higly needed. [36] The paper proposes a design and implementation of a wireless telemedicine system, in which all physiological vital signs are transmitted to remote medical server through both cellular networks for emergency cases and internet for normal cases, aiming for long-term monitoring.

A study conducted by Romano et al incorporated telemedicine with heart failure diagnostic system that uses ML. Telemedicine was used as a home monitoring instrument of patients involved. The aim of the study was to evaluate if individual patient information has an incremental informative value in automatically classifying the patient's health status rather than using physiological parameters alone. [37] Random forest classification analysis was applied to 240 complete clinical report variables in order to estimate the patient's health status; using patient's information provided via health assessment survey alone, instrumentally measured physiologal status alone,

and both taken into consideration. The study comes to a conclusion that patient monitoring is very important in providing a more correct classification of health status most especially for high risk patients that need regular health assessment.

III. Theoretical Framework

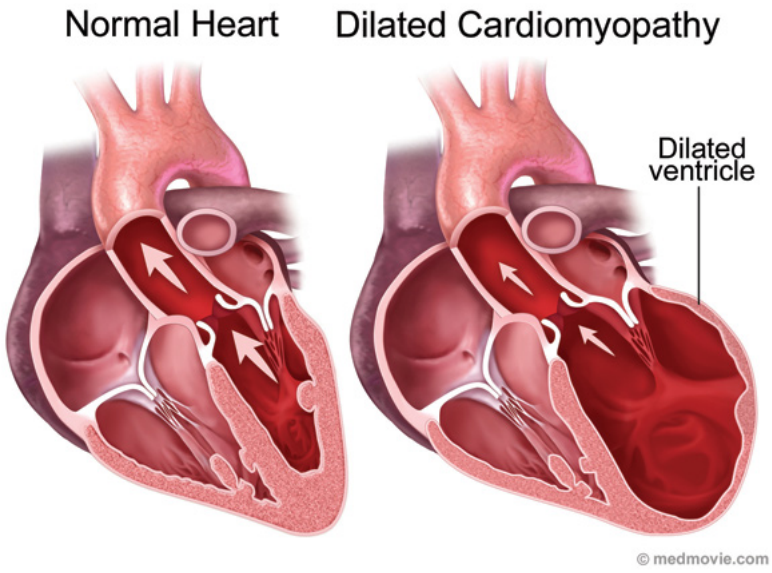
A. Cardiomegaly

Cardiomegaly is a general term used to describe any condition that results in an enlarged heart. [38]

There are two types of cardiomegaly:

1. **Dilated Cardiomyopathy**

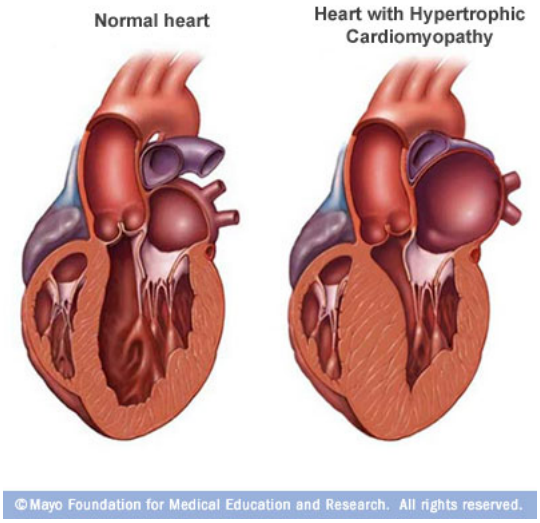
   The heart can become enlarged due to dilation of the myocardium. Dilated Cardiomyopathy (DCM) is the most common form of non-ischemic cardiomyopathy. In DCM, the heart becomes weakened and enlarged, and congestive heart failure (CHF) quickly follows. Signs and symptoms are those of left and/or right heart failure. [38]



2. **Hypertrophic Cardiomyopathy**

   Hypertrophic cardiomyopathy (HCM) is a condition where areas of heart muscle become thickened and stiff. The thickening makes it harder to contruct and pump blood out of the body. HCM generally affects the left ventricle (the main

pumping chamber), and particularly the septum (area of muscle in the middle of the heart, which separates the right and left sides). [39]



B. Chest X-Ray

A chest x ray is a fast and painless imaging test that uses electromagnetic waves to create pictures of the structures in and around your chest. This test can help diagnose and monitor conditions such as pneumonia, heart failure, lung cancer, tuberculosis, sarcoidosis, and other diseases that involve the chest area. [9]

In diagnosing cardiomegaly, the standard method for measuring heart size on CXR is known as the Danzer Method. It involves measuring the distance from the midline of the spine to the most lateral aspect of the cardiac apex (distance B in Fig 7), and adding this distance to that found from the same midline to the most lateral aspect of the right atrium (distance A). The result is then divided by the largest horizontal width (distance C), from right to left pleural surface. This value is known as the cardiothoracic ratio (CTR). A CTR > 0.5 indicates cardiomegaly. [38]

$$CTR = \frac{A + B}{C}$$

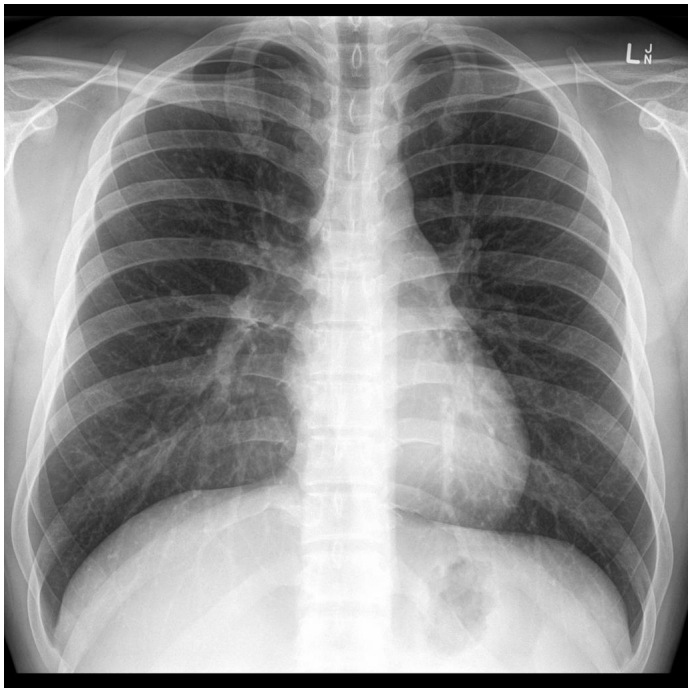


Figure 6: Normal Chest X-Ray

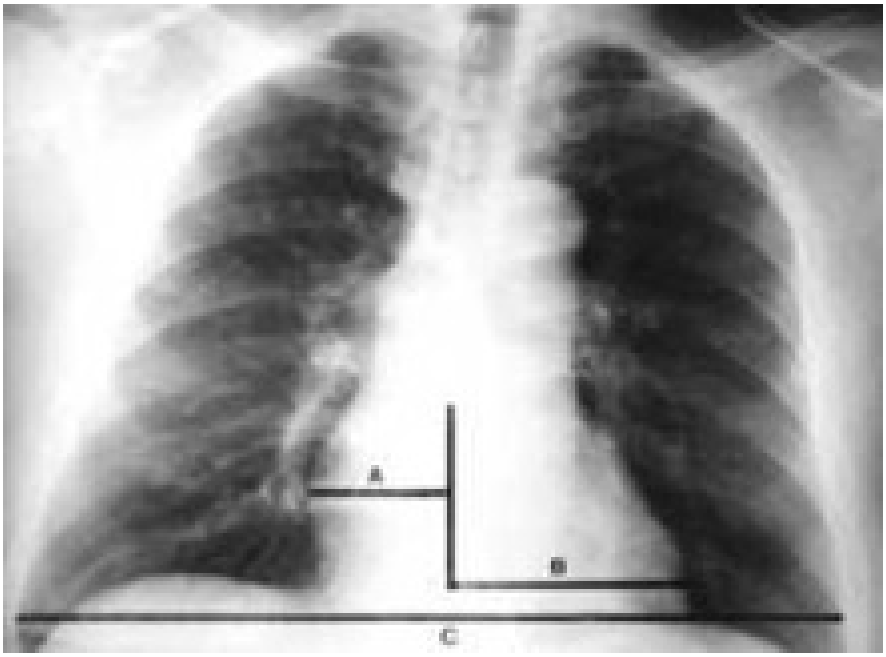


Figure 7: Measuring Cardiothoracic Ratio

C. Clinical Decision Support System

Decision Support System (DSS) is a general term for any computer application that enhances a person or group's ability to make decisions. [40] This is commonly developed for the purpose of providing solution to management problems.[22] It utilizes known data, usually built with easy-to-use interface, and allows for decision-maker's insight. In the field of healthcare, it is commonly implemented as software that helps in clinical decision-making, often used in matching the characteristics of an individual patient to a clinical knowledge base and patient-specific assessment are then presented as an output for clinical decision. [21] The main advantage of this system is adding efficiency of the healthcare providers.

D. Telemedicine

Telemedicine is the use of medical information exchanged from one site to another via electronic communications to improve a patients clinical health status. [25] It is the natural evolution of healthcare in the digital world. This includes formulating diagnosis and decision on treatment based on the data and health information transmitted via telecommunications system (eg. SMS, eMail, smart phones, video conferencing, etc.). [26] Telemedicine encompasses a wide definition of remote healthcare.Its main objective is to improve the quality, equity, and affordability of healthcare throughout every region.

E. Convolutional Neural Network

Convolutional Neural Network (CNN) is one of common DL algorithms. The goal of CNN is to learn higher-order features in the data via convolution. [17]. CNN is very well-suited to object recognition with images and many other aspect of visual data. The efficacy of CNN in image recognition is one of the main reasons for

recognizing the power of DL.

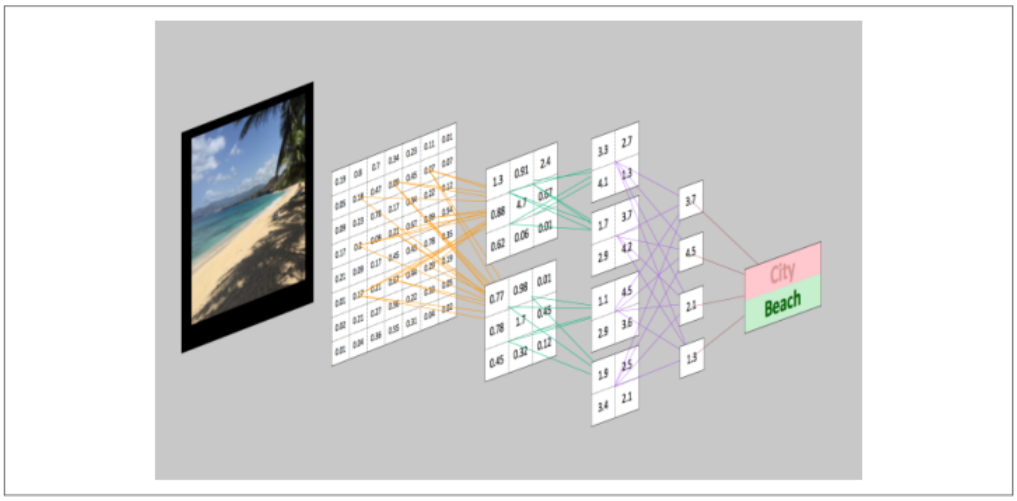


Figure 8: CNN and Computer Vision

There are many variation of CNN architectures, but they are based on the pattern of layers, as shown in Figure 9

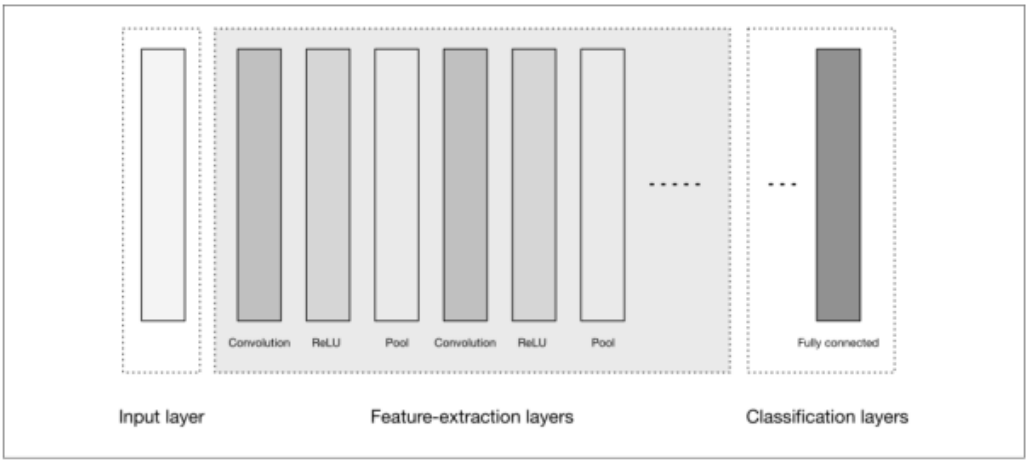


Figure 9: General CNN Architecture

CNNs transform the input data from the input layer through all connected layers into a set of class scores given by the output layer.

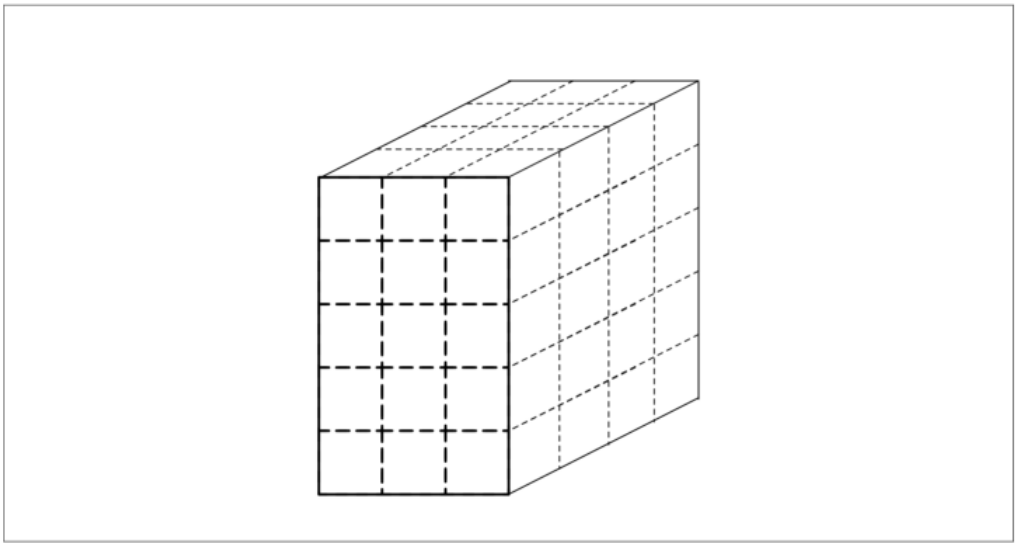


Figure 10: Input Layer

Figure 9 depicts three major groups: *Input Layer*, *Feature-Extraction Layer*, and *Classification Layer*.

The input layer accepts three-dimensional input generally in the form spatially of the size (width x height) of the image and has a depth representing the color channels.

Input layers are where we load and store the raw input data of the image for processing in the network.

The feature-extraction layers have a general repeating pattern of the sequence:

1. Convolution Layer
2. Pooling Layer

These layers find a number of features in the images and progressively construct higher-order features. This shows the advantage of deep learning by which features are automatically learned as opposed to manually-labelled features.

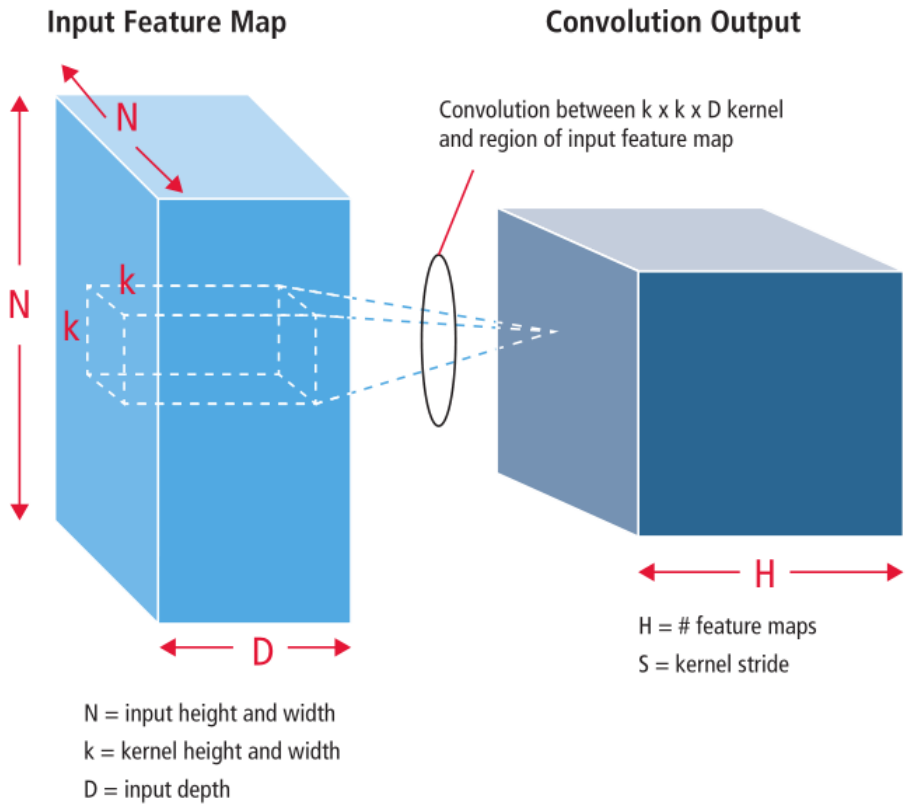


Figure 11: Convolution Layer

Convolutional layers are considered the core building blocks of CNN architectures. As Figure 11 illustrates, convolutional layers transform the input data by using a patch of locally connecting neurons from the previous layer. The layer will compute a dot product between the region of the neurons in the input layer and the weights in the kernel to which they are locally connected.

The first convolution layer extracts low-level features like edges, lines, and corners. Higher-level layers extract higher-level features.

Pooling layers are commonly inserted between successive convolutional layers. The convolutional layer is immediately followed by pooling layer to progressively reduce the spatial size of the data representation. There are two ways to do pooling: max pooling and average pooling. In both cases, the input is divided into non-

overlapping two-dimensional spaces.

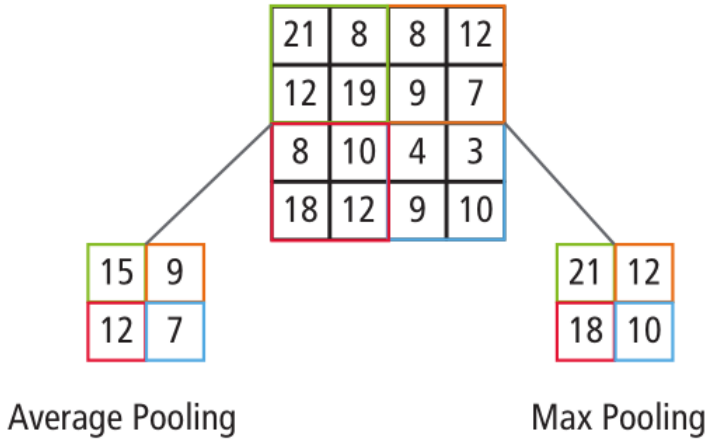


Figure 12: Pictorial Representation of max pooling ad average pooling

Figure 12 elaborates the pooling process further. The input is of size 4x4. For 2x2 subsampling, a 4x4 image is divided into four non-overlapping matrices of size 2x2. In the case of max pooling, the maximum value of the four values in the 2x2 matrix is the output. In case of average pooling, the average of the four values is the output.

Finally, the classification layer is inserted wherein one or more fully connected layers are present to take the higher-order features and produce evaluation probabilities or score of belonging in a class. These layers are fully connected to all the neurons in the previous layer.

IV. Design and Implementation

A. System Overview

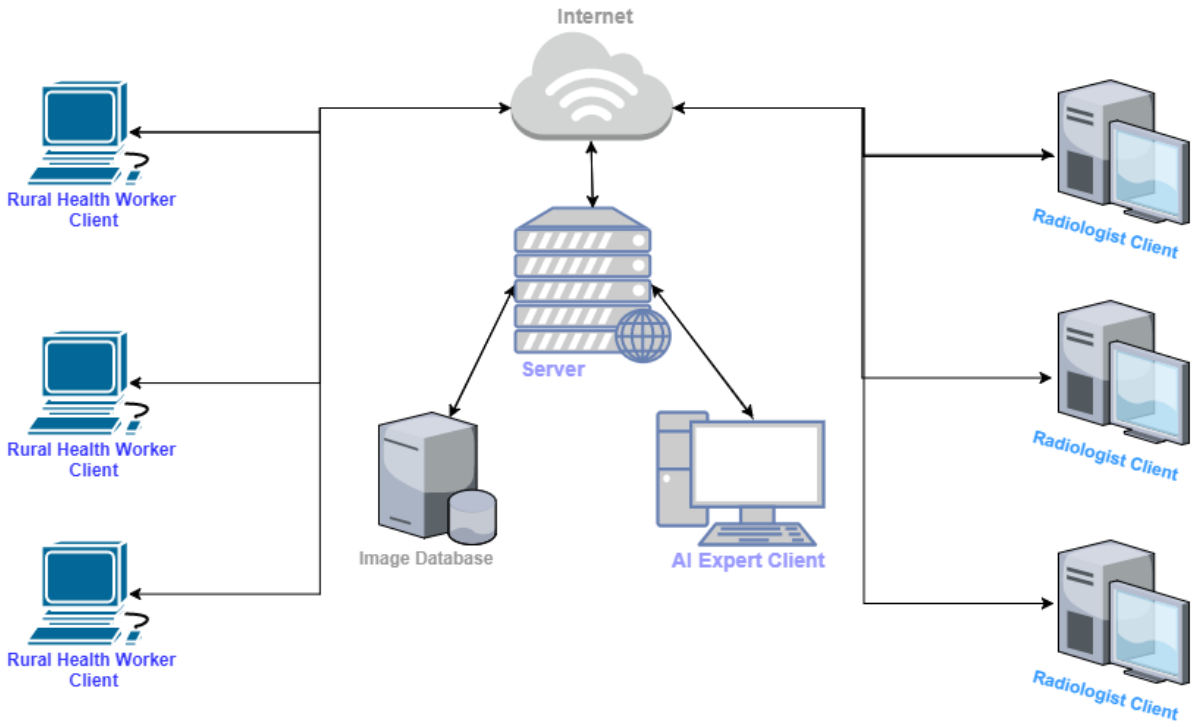


Figure 13: System Overview

The Decision Support System will be split into two parts: client and server. The client will be used to send request to the server. The server will be used to store and process the image provided by the client.

There are three types of clients:

1. Rural Health Worker Client (RHWC)
2. Radiologist Client (RC)
3. AI Expert Client (AEC)

RHWC can be accessed through a desktop computer used for sending image for classification or sending image to RC for result clarification. RC can be accessed

through a desktop computer to provide annotation to results sent by RHW. The AEC is accessed through a desktop computer used for training the model that will be used in the system.

B. Use Case Diagram

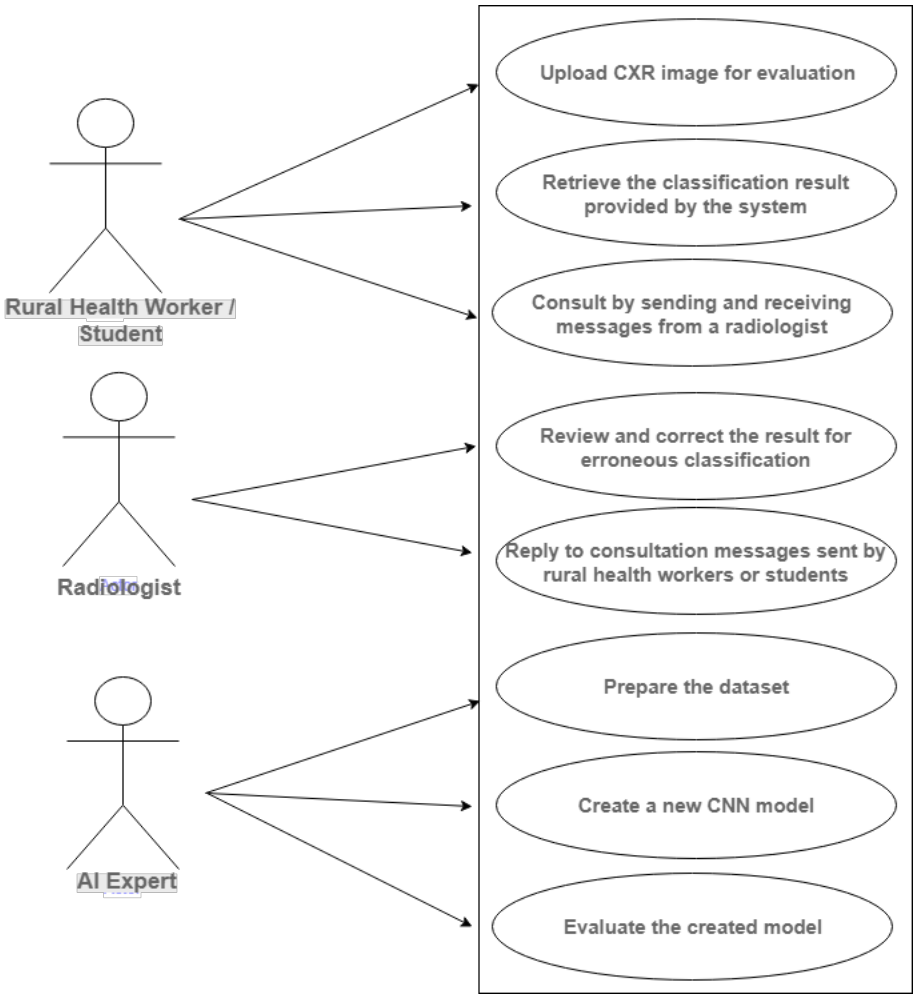


Figure 14: Use Case Diagram

Fig. 14 shows the general functionalities of the system users. The Rural Health Worker has 3 functionalities for the purpose of requesting and retrieving the classification result provided by the system, and consultation to a radiologist.

The Radiologist has a capability to review and correct diagnosis for possible

error as requested by Rural Health Worker and reply to messages and inquiries given by Rural Health Worker.

Lastly, the AI Expert has 3 functionalities: prepare the dataset to be used in training the model, create CNN model to be used by the system, and evaluation of the newly created model.

C. Entity Relationship Diagram

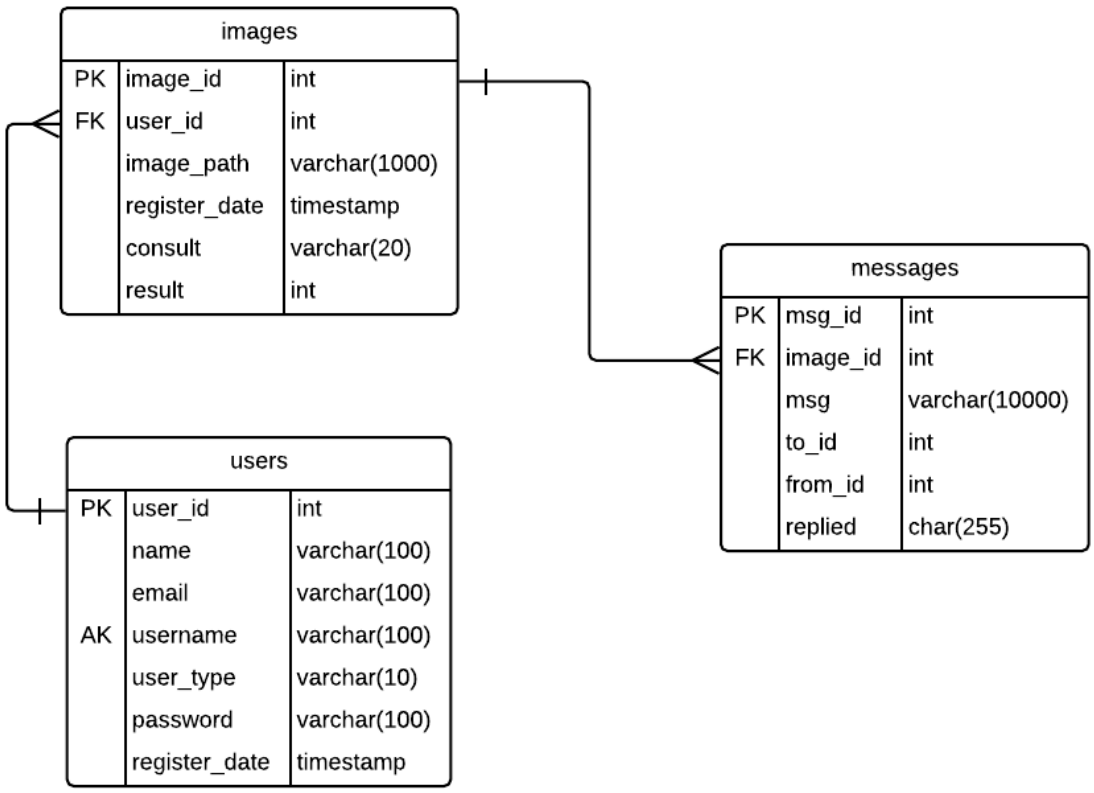


Figure 15: Entity Relationship Diagram

The system uses three tables. The first is user table. This includes unique user_id, name, email, username, sha-256-encrypted password, and register_date of the user. Next is the images table which stores the unique image_id, user_id, a foreign key

referencing the user who uploaded the image, image_path of the image in the server, register_date, the date the image has been uploaded, consult or the consultation status of the image, and result from system's prediction or annotated result of doctor. Lastly, the messages table contains the consultation message passed by the standard client and replies of doctors. This includes a unique msg_id, a foreign key image_id, msg or the message given, to_id or the receiver of the message and from_id, the sender of the message.

D. Activity Diagram

1. Rural Health Worker

The rural health worker can request a diagnosis or a diagnosis correction using the web application.

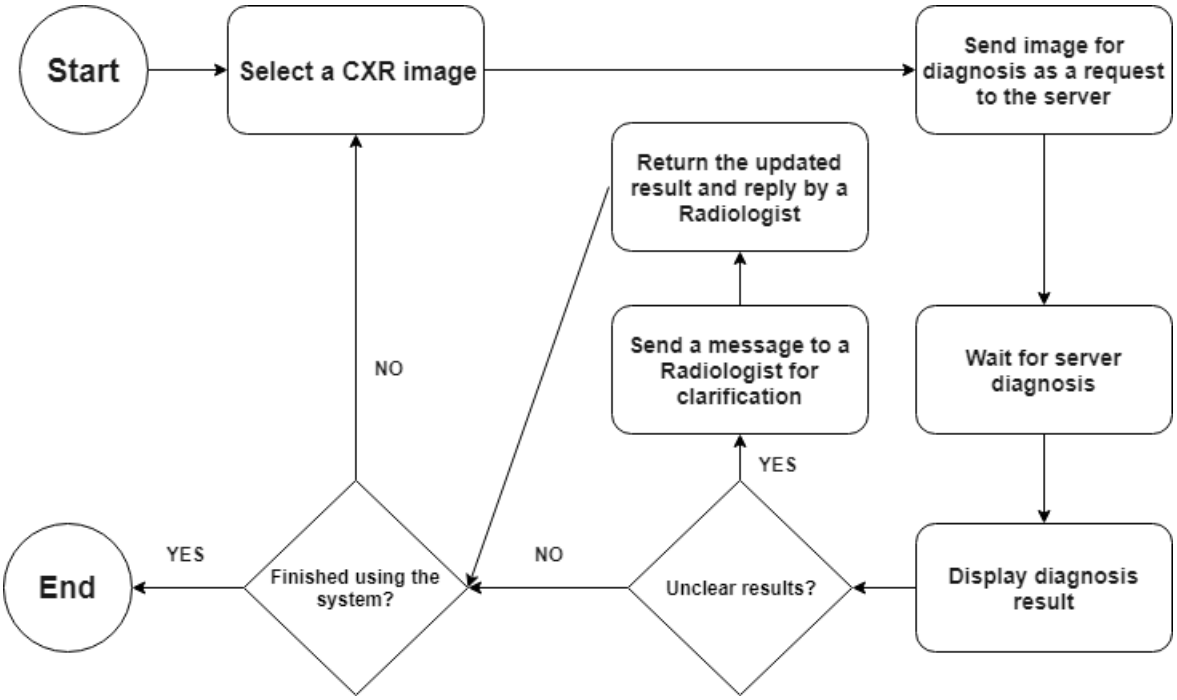


Figure 16: Rural Health Worker Activity Diagram

2. Radiologist

The correction request sent by the rural health worker will be validated by the radiologist. The corrected diagnosis will be sent back to the rural health worker and the new data will be added to the dataset.

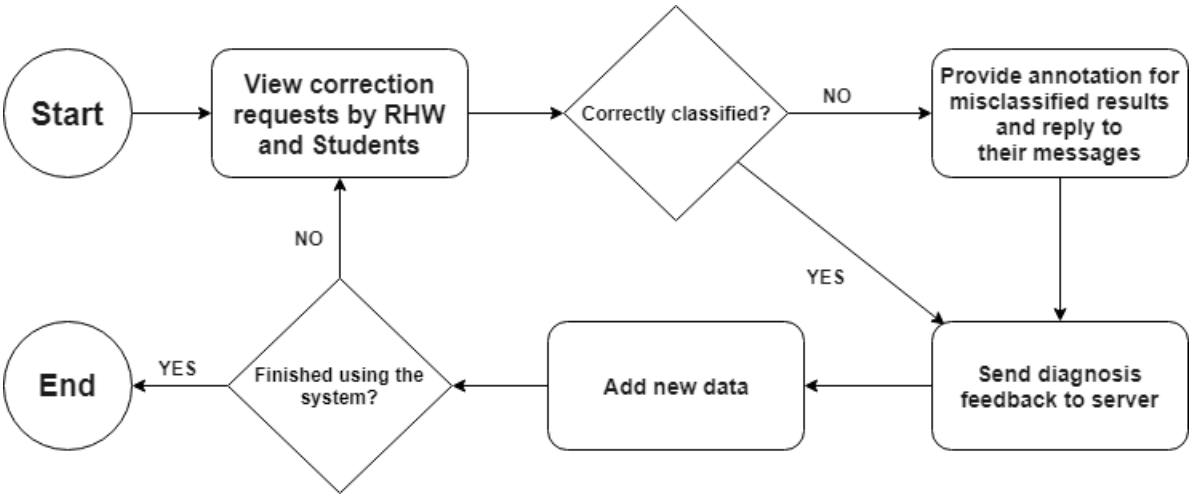


Figure 17: Radiologist Activity Diagram

3. AI Expert

The new dataset will be used by the AI expert to produce a new model that performs more accurate diagnosis. If a more suitable architecture is known, the AI expert may create a new model that uses the new architecture.

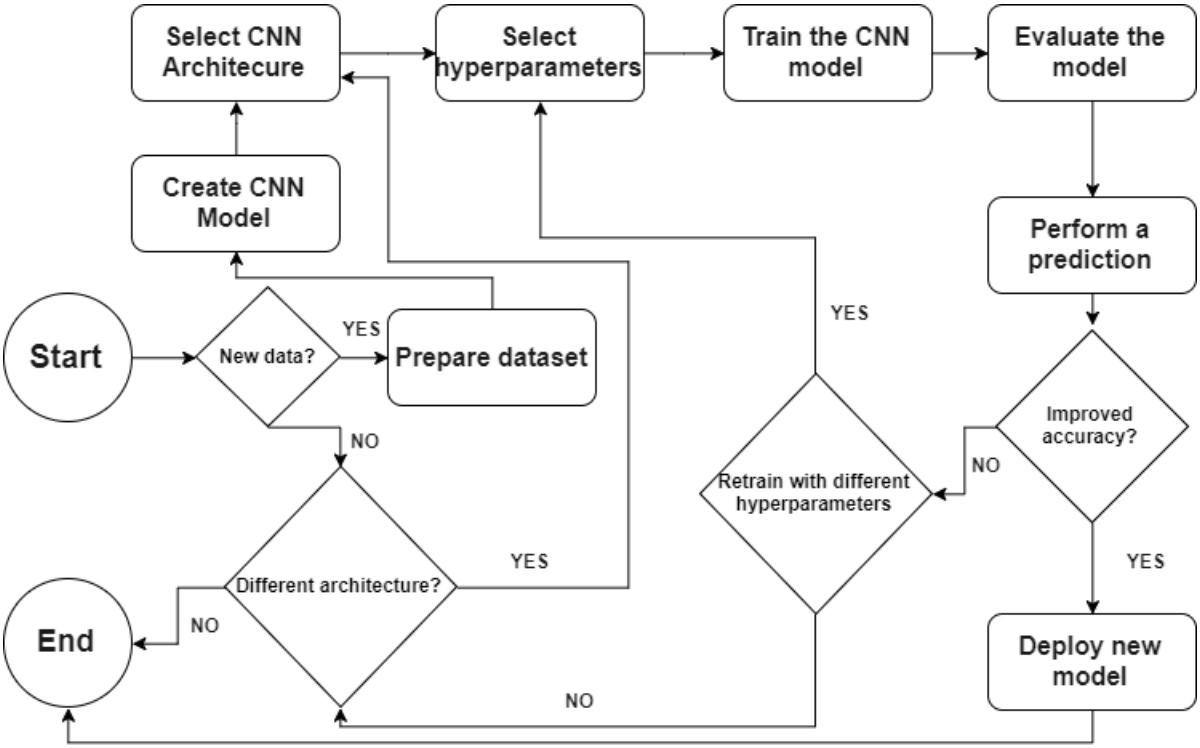


Figure 18: AI Expert Activity Diagram

E. Technical Architecture

The recommended requirements for the server machine include:

1. **2 GHz CPU rate or higher**
2. **Graphics Processing Unit (GPU)**
    (a) *NVIDIA Graphics Card with 3.0 Compute Capability or higher*
3. **8 GB RAM or higher**
4. **Up to 2 GB of free disk space**

The client side must have any of the following compatible web browsers:

1. **Mozilla Firefox**
2. **Google Chrome**
3. **Safari**

F. Dataset

The data used is a set collected from various hospitals affiliated with the Indiana University School of Medicine.[41] The set contains 7471 frontal and lateral CXRs with several abnormalities. Frontal CXR will be chosen for this study. The set is publicly available through Open-iSM [42] which is a biomedical literature search engine developed by U.S. National Library of Medicine.

V. Results

A. Home Screen

Figure 19: Homepage

As soon as the web application loads, the user is redirected to the homepage presented with functions to register or login to the system.

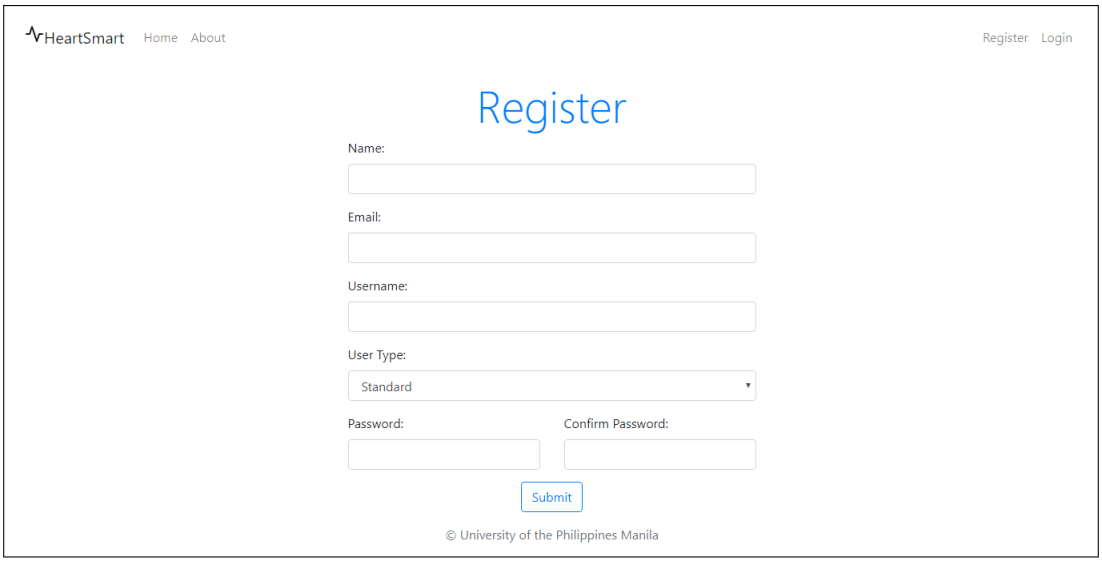


Figure 20: Register Page

New users can register and provide the necessary inputs before using the web app.

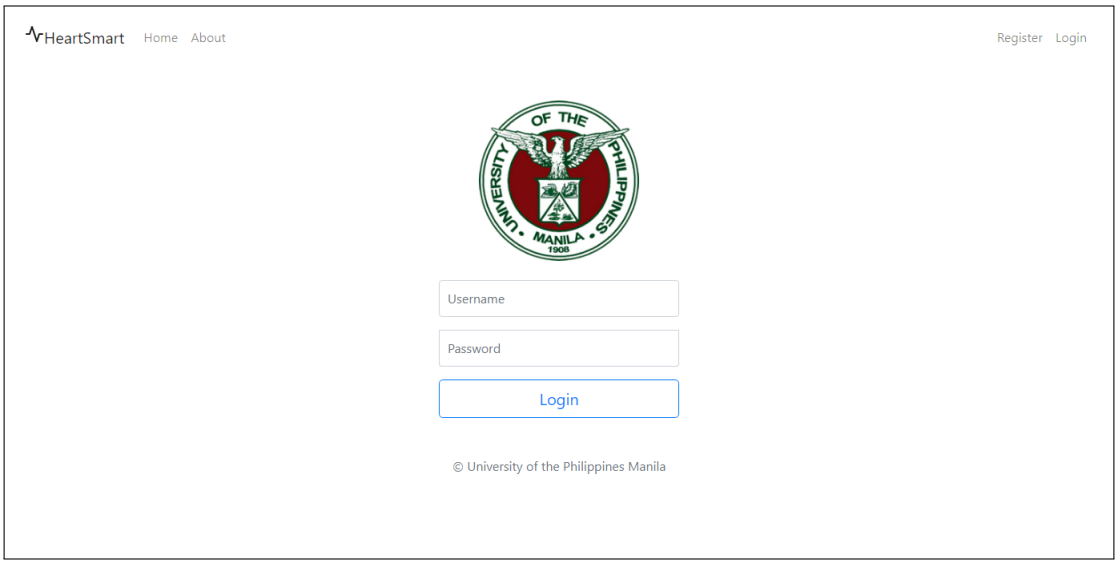


Figure 21: Login Page

Registered users can login to this page.

B. Standard User's Account

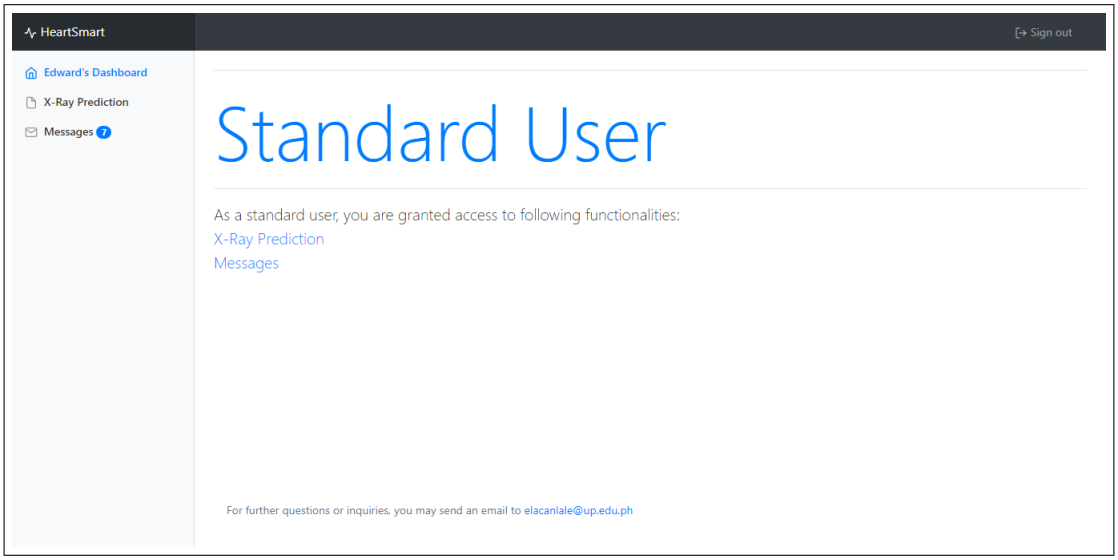


Figure 22: Standard User's Dashboard Page

Rural Health Workers and Students using the system are provided with a standard user access. Once logged-in, the dashboard shows the overview of user-allowed functionalities.

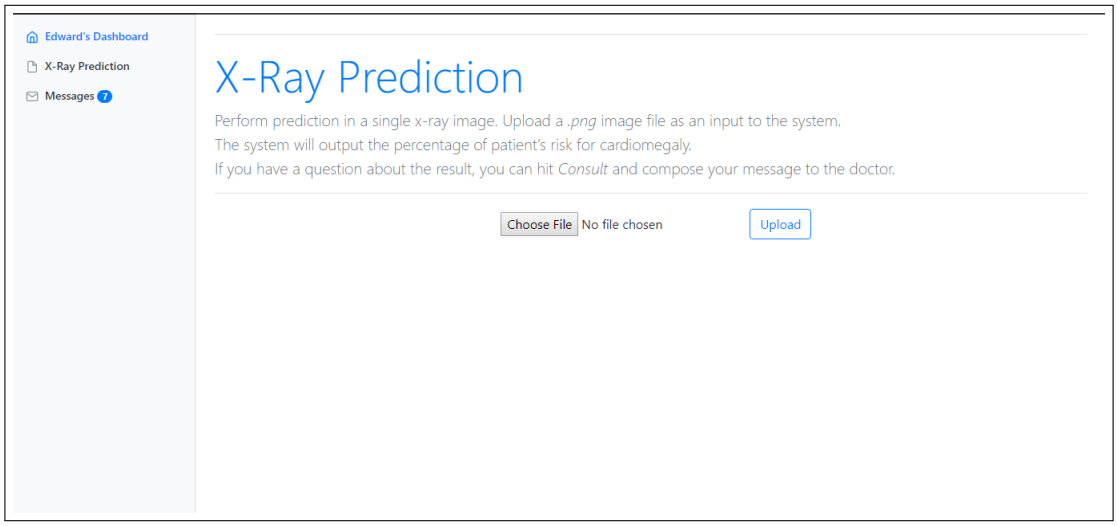


Figure 23: X-Ray Prediction Page

One of the features of the system is to predict an image. This page allows the user to provide an image for system prediction.

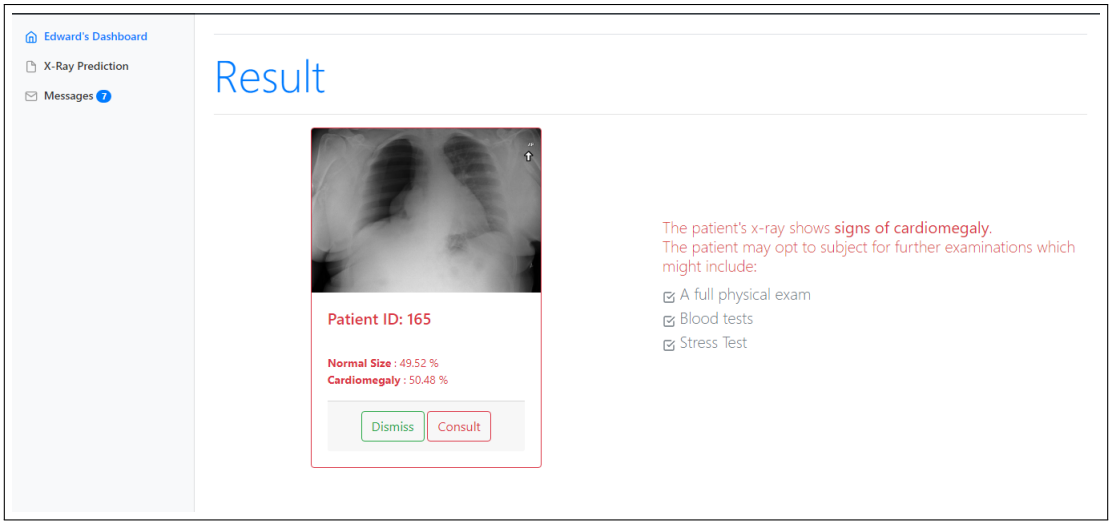


Figure 24: Prediction Result Page

This page shows the result provided by the system. The result shows the probability of belonging in the class and recommended actions are displayed. The user may accept the result outright or consult to a doctor by clicking the 'Consult' button. Suggested actions for further

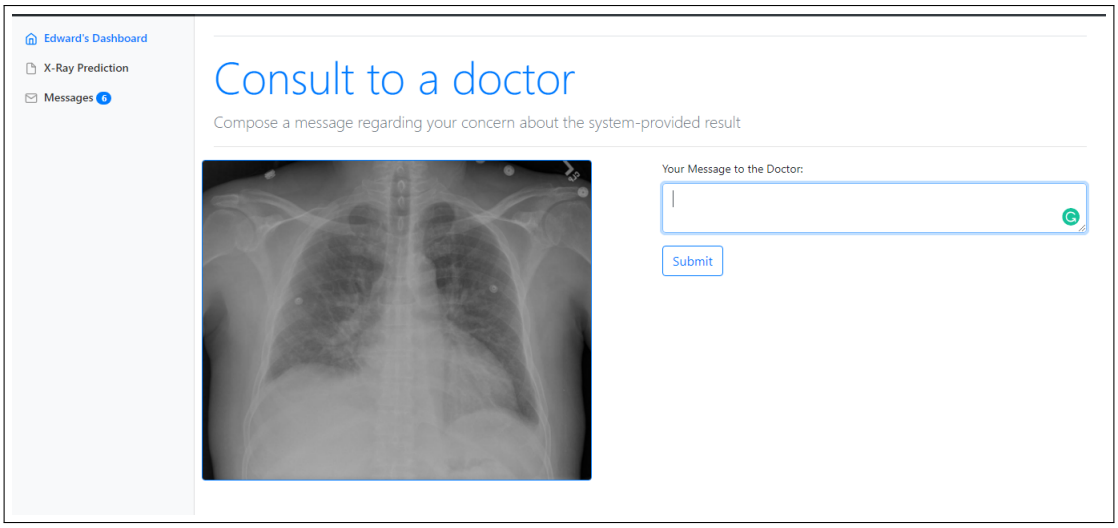


Figure 25: Consult to a Doctor Page

In the event the user chooses to consult to a doctor, the user will be redirected to a page with a form that expects message or inquiry as an input regarding the result.

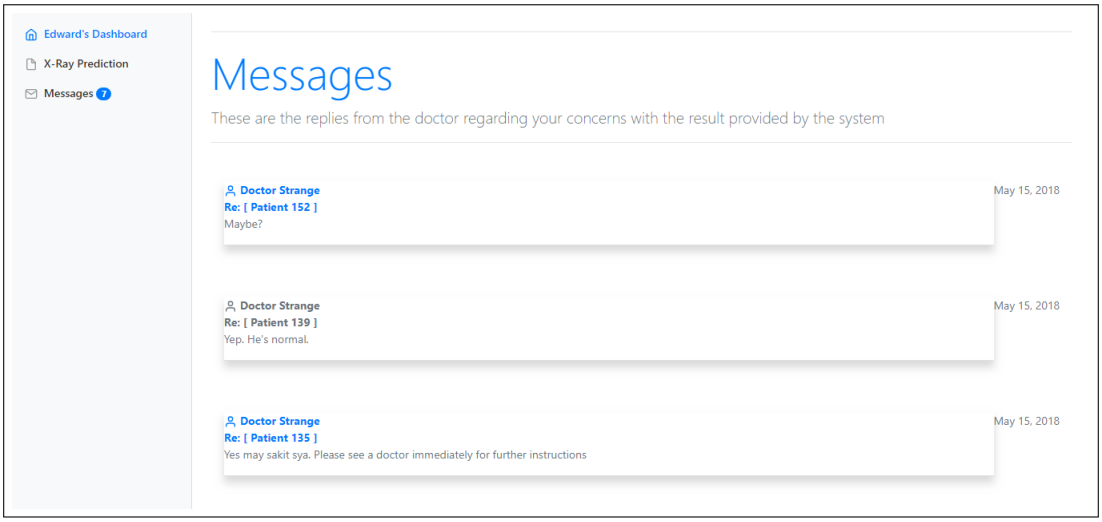


Figure 26: Message Inbox Page

This page shows the reply of the doctor regarding the consultation sent.

C. Doctor's Account

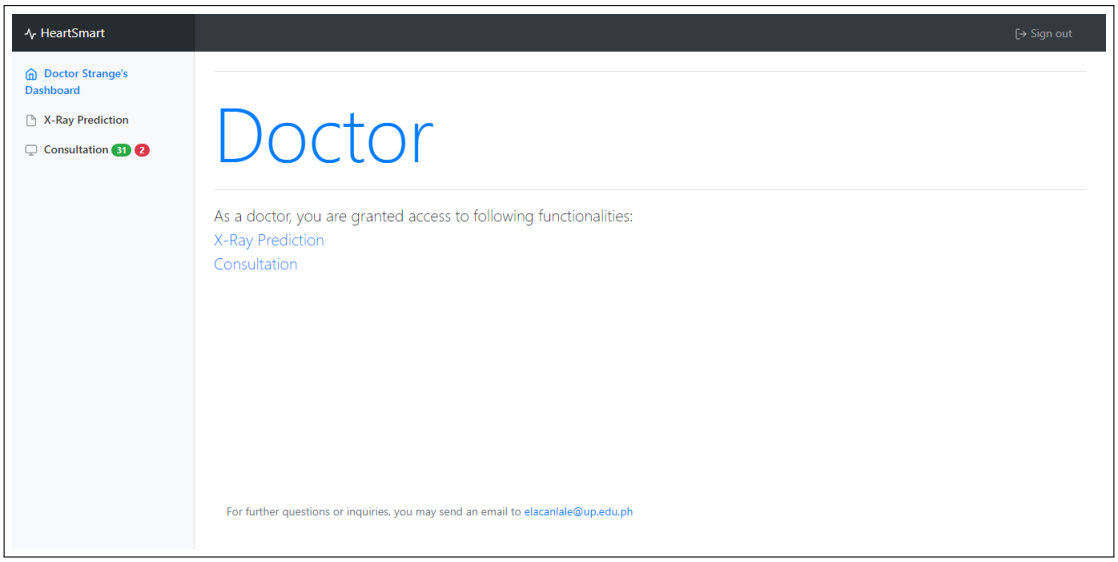


Figure 27: Doctor's Dashboard Page

Radiologists using the system are provided with a doctor user access. Once logged-in, the dashboard shows the overview of user-allowed functionalities.

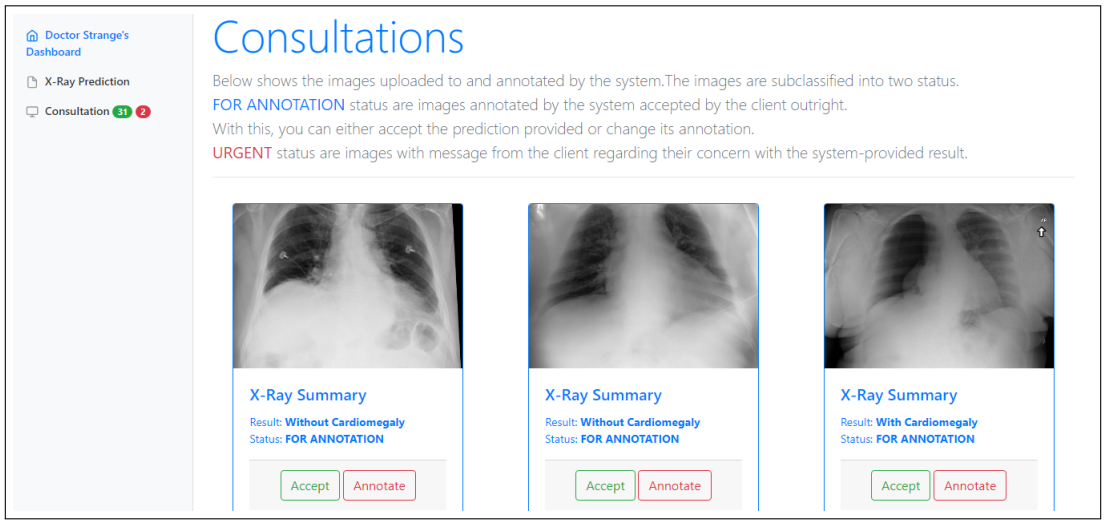


Figure 28: Consultation Page

This page shows the image input provided by other users and the system-generated prediction. The doctor has the capability to re-annotate the image or simply reply to consultation messages.

D. AI Expert's Account

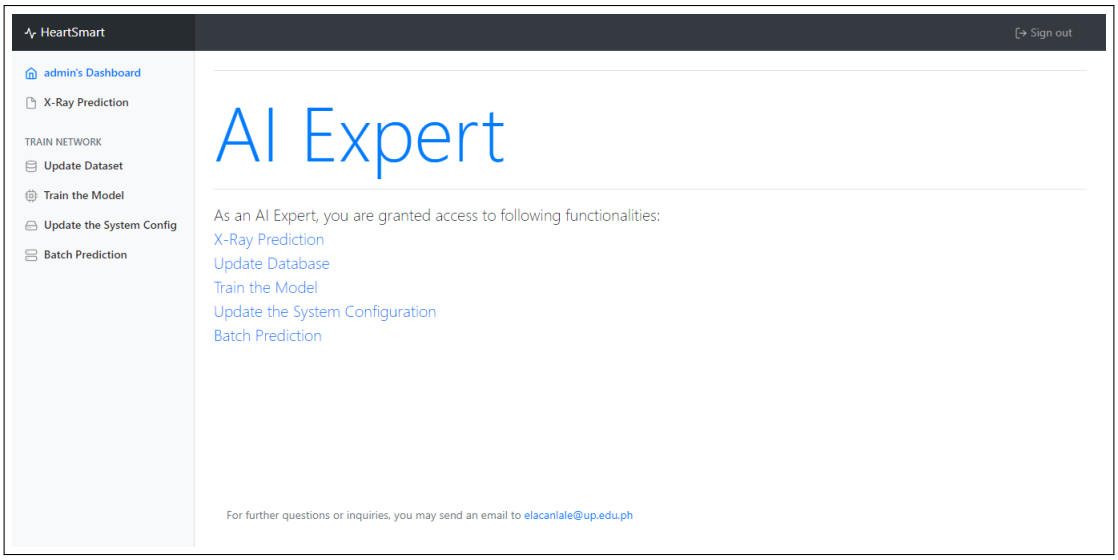


Figure 29: AI Expert's Dashboard Page

AI Expert using the system are provided with a admin user access. Once logged-in, the dashboard shows the overview of user-allowed functionalities.

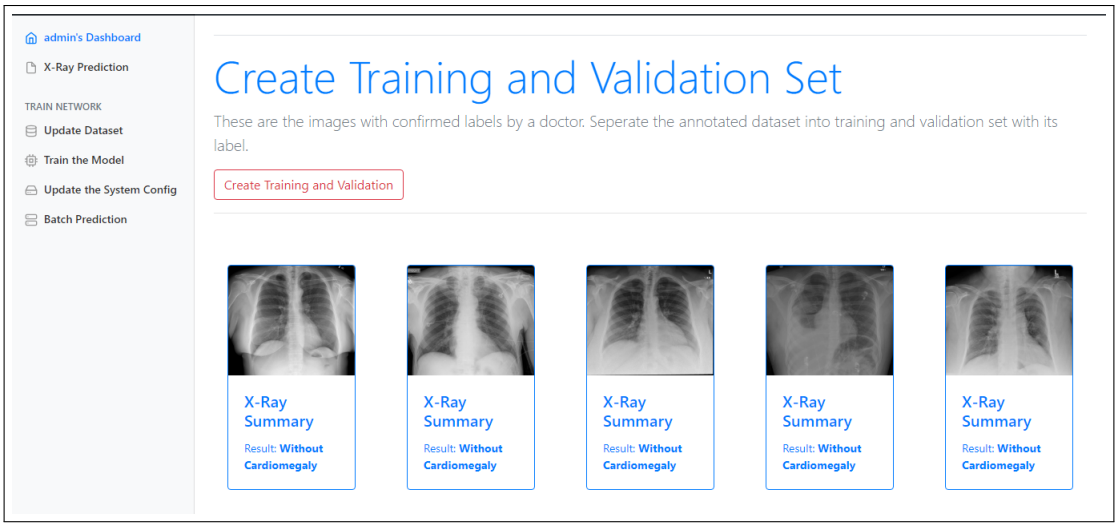


Figure 30: Create Training and Validation Set

This page shows the images with confirmed annotations from a doctor. These images may now be separated into training and validation set for model retraining.

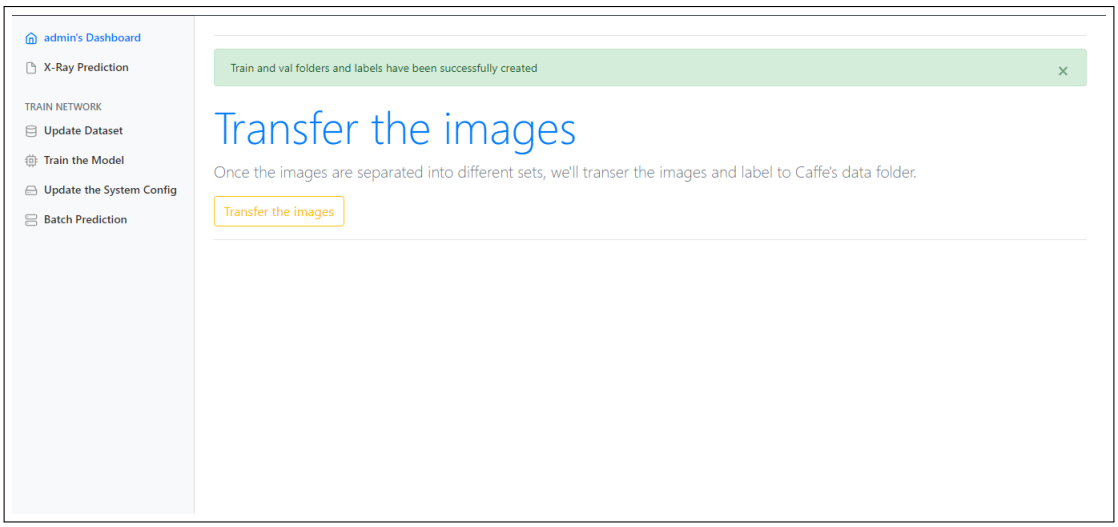


Figure 31: Transfer the Images

Once the images are properly separated, it will be transfered to Caffe's data folder.

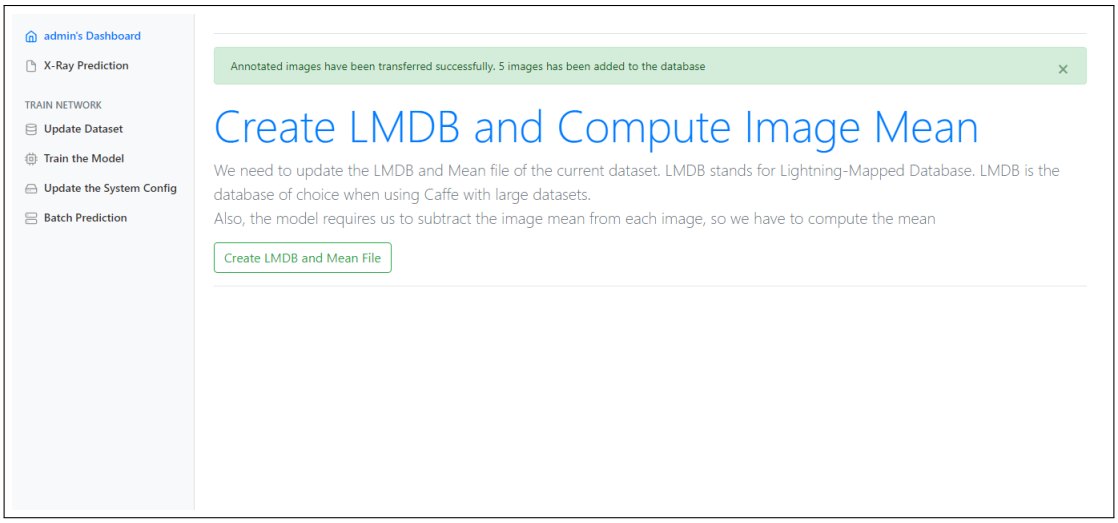


Figure 32: Create LMDB and Mean File

This page provides the capability to convert the images into LMDB and compute the images' mean.

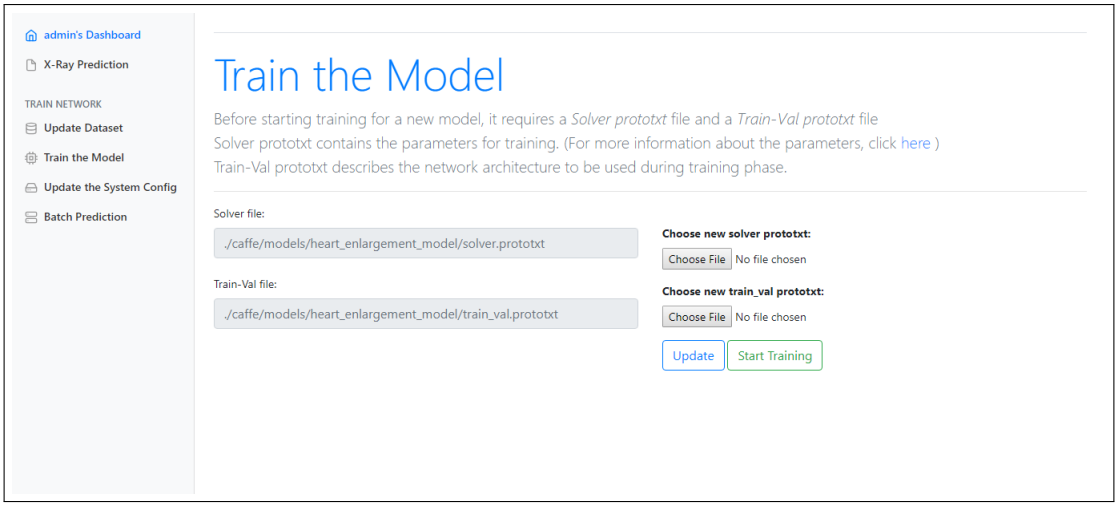


Figure 33: Train Page

This page provides the capability of the AI expert to retrain a model using user-defined solver and train-val prototxts.

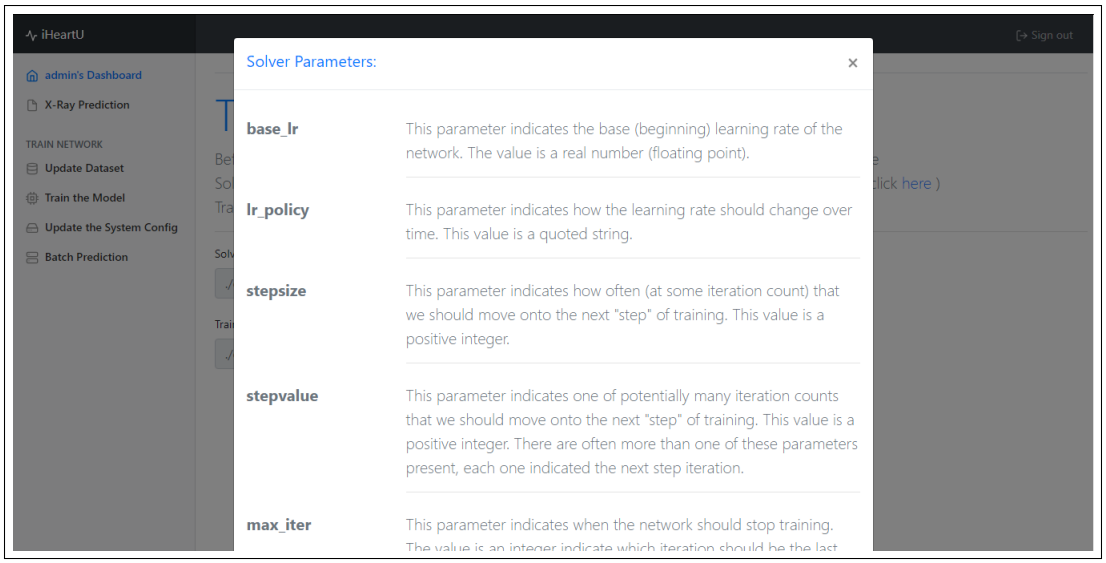


Figure 34: Solver Parameter Definitions

Solver parameter definitions are included as guide for AI Expert

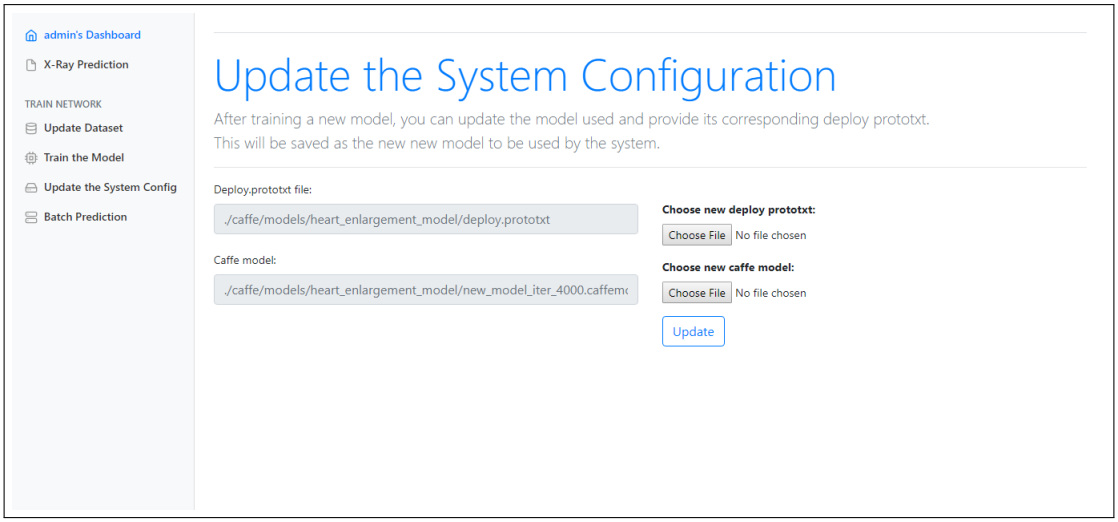


Figure 35: Update System Configuration

Once the training ended, the AI expert may choose the newly trained model to be the system's new model with its corresponding deploy prototxt.

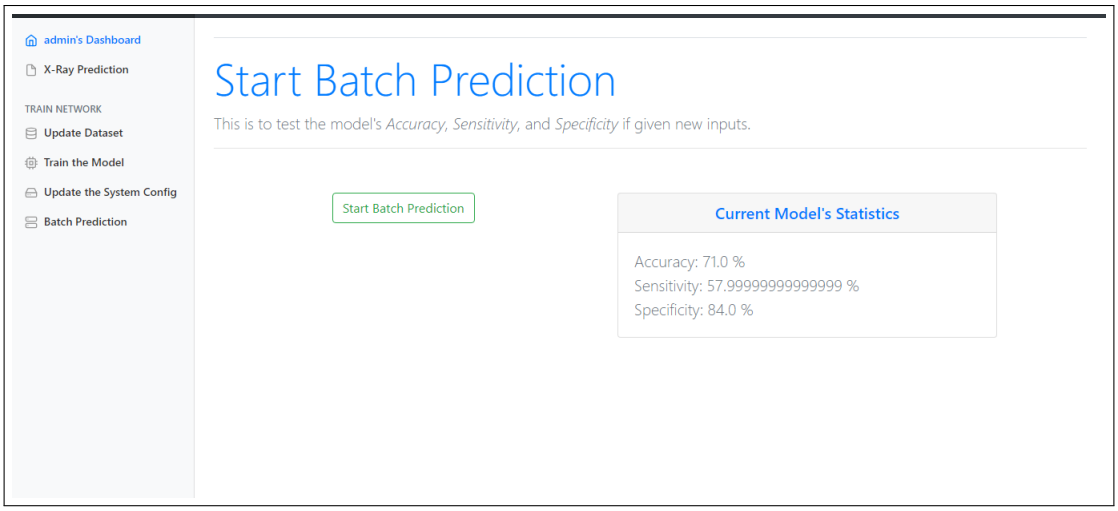


Figure 36: Batch Prediction Page

To test the predictive capability of the new model, the AI Expert may perform batch prediction. The system will compute its accuracy, sensitivity, and specificity.

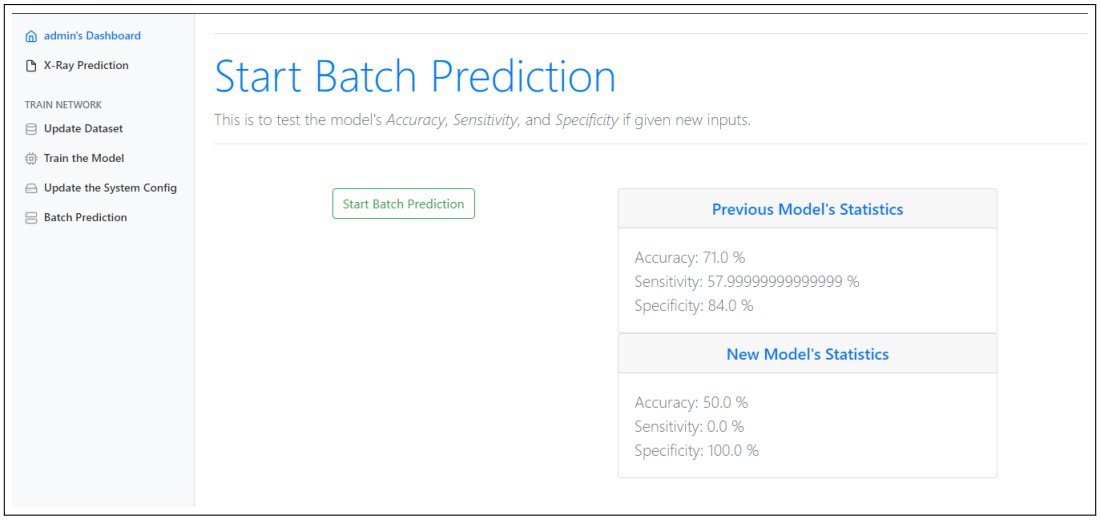


Figure 37: Model Comparison

After batch prediction, the resulting statistics from both previous and new models are displayed for AI Expert's evaluation purposes.

VI. Discussions

This cardiomegaly detection system is a web application that aims to provide a prediction of having an enlarged or normal heart given a frontal chest x-ray as input. This will serve as a decision support system to its user.

The CXR images used for training the model was from publicly available biomedical images of Indiana University School of Medicine. Frontal CXR was used that contains 3,060 normal diagnoses and 354 abnormal diagnoses. This poses a problem with class imbalance, or the trained model will highly likely produce an output leaning to the class with higher data. In this case, it will more likely predict the image to have a normal heart. To aid this problem, two possible options are available: data augmentation or data reduction. Data reduction was used for faster processing time. Data reduction involves reducing the quantity of the class with a higher number to match the amount of the class with fewer data. Both classes now have 354 images each.

Before starting the training stage, the data has to be partitioned into 60%-20%-20% training, validation, and test set. To ensure the randomness of each set, *random* was used. *random* is a python module for randomness for various statistical distribution. After ensuring data randomness, the training phase then proceeded.

Caffe framework was used in training the model with Alexnet as the CNN architecture. All parameters and layer sequences are followed except for the batch size of training and validation set, max iteration of the training process, and final class outcome. This is to fit with the problem and the dataset used. Alexnet was used due to its shallow layer sequence. This, in turn, requires lesser computational time and resources compared to other network architectures.

Flask, a python web framework, was used in the backend of the system. Javascript and CSS were used for the front-end of the web app. SQL was used for system database operations.

The model was trained using 424 images. There are 2 classes and each class consists of 212 images. For test set, 142 images are used. These are not a part of 424 images. After performing batch prediction, the model yields 71% accuracy, 54.6666% sensitivity rate or true positive rate, and 84% specificity rate or true negative rate.

VII. Conclusions

HeartSmart has been developed to give the user a web application that serves as a clinical decision support system for cardiomegaly detection. This gives the capability to rural health workers to have an immediate assessment of a patient's condition, aiding in the decision-making process. Health workers in rural areas can also use the system as a tool for teleconsultation with a specialist residing in urban hospitals. Aside for decision support system and teleconsultation, this can serve as a learning tool for radiology students for differentiating an enlarged heart from a normal one. Radiologist can use the system to respond to teleconsultations and correction of system-generated predictions.

Since the accuracy of the system is fairly average, it still needs to be improved before it can be deployed as a reliable decision support tool. Though this model is not ready for clinical adoption, this study could be used as a basis for a better decision support system for cardiomegaly detection which could later be used to provide radiologists and rural health workers with valuable information. This could later be an aid to significantly decrease the diagnosis time and improve the standard healthcare through the use of technology.

VIII. Recommendations

HeartSmart could be further improved by increasing the accuracy of the model. To further improve the accuracy of the model, it is advisable to:

1. Try other datasets with larger data.
2. Provide more images per class to the training set.
3. Try other network architectures besides AlexNet
4. Explore other deep learning methods.

The system can also be extended to cover a wider range of abnormalities found in chest x-rays.

Another feature that could be added to the system is the capability to have multiple predictions at once. The system currently allows the user to upload one image at a time. This could lessen the waiting time of each user.

Lastly, a mobile version of HeartSmart can be developed for convenience and utilize the increasing number of mobile users in both urban and rural areas.

IX. Bibliography

[1] "Mortality." http://www.doh.gov.ph/mortality. Accessed: 2017-09-04.

[2] "What is cardiovascular disease?." http://www.heart.org/HEARTORG/Conditions/What-is-Cardiovascular-Disease_UCM_301852_Article.jsp#.Wblph8gjHDc. Accessed: 2017-09-04.

[3] "Cardiovascular disease is still the country's top killer." http://www.pchrd.dost.gov.ph/index.php/news/library-health-news/4123-cardiovascular-disease-is-still-the-country-s-top-killer. Accessed: 2017-09-04.

[4] "What is coronary heart disease?." https://www.nhlbi.nih.gov/health/health-topics/topics/cad. Accessed: 2017-09-04.

[5] "Cardiomegaly." https://radiopaedia.org/articles/cardiomegaly. Accessed: 2017-09-27.

[6] P. Ponikowski, , A. A. Voors, , S. D. Anker, , and H. e. a. Bueno, "2016 esc guidelines for the diagnosis and treatment of acute and chronic heart failure. the task force for the diagnosis and treatment of acute and chronic heart failure of the european society of cardiology (esc)developed with the special contribution of the heart failure association (hfa) of the esc," *European Heart Journal*, vol. 37, no. 27, pp. 2129–2200, 2016.

[7] "What is heart failure." http://www.heart.org/HEARTORG/Conditions/HeartFailure/AboutHeartFailure/What-is-Heart-Failure_UCM_002044_Article.jsp#.WhLs6jdx3MU. Accessed: 2017-11-20.

[8] A. B. Monfared, S. A. Farajollah, F. Sabour, R. Farzanegan, and S. Taghdisi, "Comparison of radiological findings of chest x-ray with echocardiography in

determination of the heart size," *Iranian Red Crescent Medical Journal*, January 2015.

[9] "Chest x-ray." `https://www.nhlbi.nih.gov/health/health-topics/topics/cxray`. Accessed: 2017-11-26.

[10] A. Brady, R. O. Laoide, P. McCarthy, and R. McDermont, "Discrepancy and error in radiology: Concepts, causes and consequences," *Ulster Med Journal*, 2012.

[11] F. S. R. F. S. T. Ali Biharas Monfared, Shahnaz Agha Farajollah, "Comparison of radiological findings of chest x-ray with echocardiography in determination of the heart size," *Iranian Red Crescent Medical Journal*, vol. 17, January 2015.

[12] P. Gupta, "Machine learning: The future of healthcare." `https://harvardsciencereview.com/2017/05/16/machine-learning-the-future-of-healthcare/`. Accessed: 2017-10-08.

[13] T. Saloky and J. eminsk, "Artificial intelligence and machine learning," *Virtual Program Modules of AI Systems*, 2005.

[14] M. A. Kutlay and S. Gagula-Palalic, "Application of machine learning in healthcare: Analysis on mhealth dataset," *Southeast Europe Journal of Soft Computing*, September 2015.

[15] N. Buduma and N. Locascio, *Fundamentals of Deep Learning*. O'Reilly Media, Inc., 2017.

[16] A. C. Ian Goodfellow, Yoshua Bengio, *Deep Learning*. MIT Press, 2016.

[17] J. Patterson and A. Gibson, *Deep Learning: A Practitioner's Approach*. O'Reilly Media, Inc, August 2017.

[18] "Seeking a better way to find web images." http://www.nytimes.com/2012/11/20/science/for-web-images-creating-new-technology-to-seek-and-find.html?_r=0. Accessed: 2017-11-28.

[19] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[20] "Ilsvrc 2017 overview." http://image-net.org/challenges/talks_2017/ILSVRC2017_overview.pdf. Accessed: 2017-11-28.

[21] C. Castaneda, K. Nalley, C. Mannion, P. Bhattacharyya, P. Blake, A. Pecora, A. Goy, and K. S. Suh, "Clinical decision support systems for improving diagnostic accuracy and achieving precision medicine," *Journal of Clinical Bioinformatics*, March 2015.

[22] D. J. Power, *Decision Support Systems: Frequently Asked Questions*. iUniverse, 2004.

[23] I. C. Gonzales, "Teleradiology provider aims to widen reach in rural areas," *The Philippine Star*, Mar 2015.

[24] "Congress proposes law on telehealth, up nthc as resource." https://telehealth.ph/2015/01/28/congress-files-law-on-telehealth-up-nthc-as-resource/. Accessed: 2017-10-06.

[25] "The ultimate frontier for superior healthcare delivery." http://www.americantelemed.org/about/about-telemedicine. Accessed: 2017-10-06.

[26] R. Henri Joshua Igna, "Infographic: How does telemedicine work in nthc?." https://telehealth.ph/2015/02/18/infographic-how-does-telemedicine-work-in-nthc/. Accessed: 2017-10-06.

[27] J. M. C. de Gea, G. Garca-Mateos, J. L. Fernndez-Alemn, and J. L. Hernndez-Hernndez2, "A computer-aided detection system for digital chest radiographs," *Journal of Healthcare Engineering*, February 2016.

[28] Y. Bar, I. Diamant, L. Wolf, S. Lieberman, E. Konen, and H. Greenspan, "Chest pathology detection using deep learning with non-medical training," *Computer Methods in Biomechanics and Biomedical Engineering: Imaging and Visualization*, April 2015.

[29] C. Tataru, D. Yi, A. Shenoyas, and A. Ma, "Deep learning for abnormality detection in chest x-ray images," June 2017.

[30] K. Z. B. Y. H. M. T. D. D. D. A. B. C. L. K. S. M. P. L. A. Y. N. Pranav Rajpurkar, Jeremy Irvin, "Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning," November 2017.

[31] M. T. Islam, M. A. Aowal, A. T. Minhaz, and K. Ashraf, "Abnormality detection and localization in chest x-rays using deep convolutional neural networks," *CoRR*, May 2017.

[32] W. Dai, J. Doyle, X. Liang, H. Zhang, N. Dong, Y. Li, and E. P. Xing, "Scan: Structure correcting adversarial network for chest x-rays organ segmentation," *CoRR*, vol. abs/1703.08770, 2017.

[33] M. Avendi, A. Kheradvar, and H. Jafarkhani, "A combined deep-learning and deformable-model approach to fully automatic segmentation of the left ventricle in cardiac mri," *Medical image Analysis*, vol. 30, August 2015.

[34] L. Zhao and K. Jia, "Multiscale cnns for brain tumor segmentation and diagnosis," *Computational and Mathematical Methods in Medicine*, October 2015.

[35] T. McKechnie, "Telemedicine: The emergence of technology in healthcare practices and the resulting implications on patient safety and healthcare efficiency," *QIHI Journal of Healthcare Improvement and Patient Safety*.

[36] M. Abo-Zahhad, S. M. Ahmed, and O. Elnahas, "A wireless emergency telemedicine system for patients monitoring and diagnosis," *International Journal of Telemedicine and Applications*, October 2013.

[37] M. Maria Francesca Romano, M. Maria Vittoria Sardella, M. Frabrizio Alboni, M. Antonio L'Abbate, M. Rita Mariotti, and M. Vitantonio Di Bello, "The informative contribution of the "virtual medical visit" in a new heart failure telemedicine integrated system," *Telemedicine and e-Health*, June 2014.

[38] "Clinical cases: Enlarged heart (cardiomegaly)." `https://www.med.umich.edu/lrc/coursepages/m1/anatomy2010/html/clinicalcases/cardiomegaly/cardiomegaly.html`. Accessed: 2017-10-25.

[39] "Hypertrophic cardiomyopathy (hcm)." `http://www.cardiomyopathy.org/hypertrophic-cardiomyopathy/intro`. Accessed: 2017-10-25.

[40] J. N. J. R. G. R. W. a. D. S. D. P. F. B. D. J. P. R. S. e. Paul Gray, Bob Johansen, *Decision Support: An Examination of the DSS Discipline*. Annals of Information Systems 14, Springer-Verlag New York, 2011.

[41] S. Candemir, S. Jaeger, W. Lin, Z. Xue, S. Antani, and G. Thoma, "Automatic heart localization and radiographic index computation in chest x-rays," in *Medical Imaging 2016: Computer-Aided Diagnosis*, vol. 9785, March 2016.

[42] "National library of medicine." https://openi.nlm.nih.gov//. Accessed: 2017-10-25.

X. Appendix

A. Source Code

app.py

```
import os
import datetime
import utilities
import numpy as np

from functools import wraps
from flask_mysqldb import MySQL
from passlib.hash import sha256_crypt
from werkzeug.utils import secure_filename
from _mysql_exceptions import IntegrityError
from subprocess import check_output, call, Popen, PIPE
from flask import Flask, render_template as rt, request
from flask import flash, redirect, url_for, session, logging
from flask_uploads import UploadSet, configure_uploads,
    IMAGES, ALL
from wtforms import Form, StringField, PasswordField,
    validators, SelectField, TextAreaField

app = Flask(__name__)
app.debug = True

# Config MySQL
app.config['MYSQL_HOST'] = 'localhost'
app.config['MYSQL_USER'] = 'edward'
app.config['MYSQL_PASSWORD'] = 'edward'
app.config['MYSQL_DB'] = 'heartdb'
app.config['MYSQL_CURSORCLASS'] = 'DictCursor'
# Initialize MySQL
mysql = MySQL(app)

# Initialize image uploads
photos = UploadSet('photos', IMAGES)
app.config['UPLOADED_PHOTOS_DEST'] = 'static/img'
configure_uploads(app, photos)

# Initialize Caffe files uploads
model = UploadSet('model', extensions=('prototxt', 'caffemodel
    '))
app.config['UPLOADED_MODEL_DEST'] = 'caffe/models/
    heart_enlargement_model'
configure_uploads(app, model)

@app.route('/')
def index():
    return rt('home.html')

@app.route('/about')
def about():
    return rt('about.html')

class RegisterForm(Form):
    name = StringField('Name', [
        validators.DataRequired(),
        validators.Length(min=1, max=50)
        ])
    username = StringField('Username', [
        validators.DataRequired(),
        validators.Length(min=1, max=50)
        ])
    user_type = SelectField(u'User Type',
        choices = [ ('STANDARD', 'Standard'), ('DOCTOR', '
            Doctor')]
        )
    email = StringField('Email', [validators.Length(min=6,
        max=50)])
    password = PasswordField('Password', [
        validators.DataRequired(),
        validators.EqualTo('confirm', message='Password do
            not match')
    ])
    confirm = PasswordField('Confirm Password')

@app.route('/register', methods=['GET', 'POST'])
def register():
    form = RegisterForm(request.form)
    userTypes = ['ADMIN', 'DOCTOR', 'STANDARD']

    if request.method == 'POST' and form.validate():
        name = form.name.data
        email = form.email.data
        username = form.username.data
        user_type = form.user_type.data
        password = sha256_crypt.encrypt(str(form.password.data
            ))
        #app.logger.info(form.user_type.data)

        # Create cursor
        cur = mysql.connection.cursor()
        try:
            cur.execute("INSERT INTO users(name, email,
                username, user_type, password) VALUES(%s,
                %s, %s, %s, %s)", (name, email, username,
                user_type, password))
        except IntegrityError as IE:
            print(cur)
            flash('Sorry, This user is already registered ', '
                danger')
            rt('register.html', form=form)
        mysql.connection.commit()
        cur.close()
        flash('You successfully registered ', 'success')

        return redirect(url_for('index'))

    return rt('register.html', form=form)

@app.route('/login', methods=['GET', 'POST'])
def login():
    #print(request.method)
    #print(request)
    if request.method == 'POST':
        username = request.form['username']
        password_candidate = request.form['password']
        #Create a cursor
        cur = mysql.connection.cursor()

        result = cur.execute("SELECT * FROM users WHERE
            username = %s", [username])
        #print(result)
        if result > 0:
            #Get stored hash
            data = cur.fetchone()
            password = data['password']

            #Compare password
            if sha256_crypt.verify(password_candidate,
                password):
                #app.logger.info('PASSWORD MATCHED')
                session['logged_in'] = True
                session['name'] = data['name']
                session['user_type'] = str.lower(data['user_type
                    '])
                session['user_id'] = int(data['user_id'])

                flash('You are now logged in', 'success')

                return redirect(url_for('dashboard'))
            else:
                error = "Invalid password"
                flash("Invalid Password", "danger")
                return rt('login.html', error = error)

            cur.close()
        else:
            error = "Username not found"
            flash("Username not found", "danger")
            return rt('login.html', error = error)

    return rt('login.html')

def is_logged_in(f):
    @wraps(f)
    def wrap(*args, **kwargs):
        if 'logged_in' in session:
            return f(*args, **kwargs)
        else:
            flash("Please login", 'danger')
            return redirect(url_for('login'))
    return wrap


@app.route('/dashboard')
```

```
@is_logged_in
def dashboard():
    cur = mysql.connection.cursor()
    cur.execute("SELECT image_path FROM images WHERE
        consult='FOR ANNOTATION' ")
    session['consult'] = cur.fetchall()
    cur.execute("SELECT image_path FROM images WHERE
        consult='URGENT' ")
    session['urgent'] = cur.fetchall()
    if session['user_type'] == "standard":
  query = "SELECT * FROM messages WHERE to_id=%s"
        cur.execute(query, (session['user_id'],) )
    else:
 query = "SELECT * FROM messages WHERE replied = %s
      AND (to_id=%s OR to_id=%s)"
        cur.execute(query, ('N',session['user_id'],0,) )
    session['messages']=cur.fetchall()
    cur.close()
    if session['user_type'] == 'standard':
        return rt('dashboard.html')
    elif session['user_type'] == 'doctor':
        return rt('dashboard-doctor.html')
    else:
        return rt('dashboard-admin.html')

@app.route('/upload', methods = ['GET', 'POST'])
@is_logged_in
def upload():
    cur = mysql.connection.cursor()
    imgs = cur.execute("SELECT * from images")
    newName = str(imgs)+'.png'

    cur.close()
    if request.method == 'POST' and 'photo' in request.files:
        photos.save(request.files['photo'], name = newName)
        session['current_img'] = os.path.normpath(os.path.join(
            os.getcwd()+'/'
  +app.config['UPLOADED_PHOTOS_DEST']+'/'+newName)).
        replace('\\', '/')
        session['temp'] = '/'+app.config['
            UPLOADED_PHOTOS_DEST']+'/'+newName
        flash("File uploaded", 'success')
        return redirect(url_for('predict'))
    elif request.method == 'POST' and 'photo' not in request.
        files:
        flash('No file selected', 'danger')
    return rt('upload.html', card = False, img = '')

@app.route('/predict', methods=['GET', 'POST'])
def predict():
    cur = mysql.connection.cursor()
    json_config = utilities.read_config_file()
    MEAN_FILE = json_config['mean']
    DEPLOY = json_config['deploy']
    WEIGHTS = json_config['weights']
    CLASSIFY = json_config['classify']
    RESULT = json_config['result']

    session['result'] = check_output(["python", CLASSIFY,
        session['current_img'], RESULT, "--pretrained_model
        ", WEIGHTS, "--model_def", DEPLOY, "--
        mean_file", MEAN_FILE, "--ext", "png", "--
        center_only"]).decode("utf-8")

    cur.execute('INSERT INTO images(user_id, image_path,
        result) values(%s, %s, %s)',(session['user_id'], session
        ['temp'], int(session['result'])))
    mysql.connection.commit()
    cur.execute('SELECT image_id FROM images where
        image_path = %s', (session['temp'],))
    patient_id = cur.fetchone()['image_id']
    cur.close()

    prob = np.load('./static/result.npy')

    return rt('upload.html', card = True, prob=prob[0], pid =
        patient_id)

@app.route('/users')
@is_logged_in
def users():
    cur = mysql.connection.cursor()
    cur.execute("SELECT * FROM users")
    users = cur.fetchall()

    if len(users)>0:
        return rt('users.html', users = users)

@app.route('/summary')
@is_logged_in
def summary():
    cur = mysql.connection.cursor()
    if session['user_type'] == 'standard':
        result = cur.execute("SELECT image_path FROM
            images WHERE user_id=%s", (session['user_id'],))
    else:
        result = cur.execute("SELECT image_path FROM
            images")
    img_paths = cur.fetchall()
    #print(img_paths)
    return rt('summary.html', img_paths=img_paths)

@app.route('/consultation')
@is_logged_in
def consultation():

    if request.args.get('path'):
        cur = mysql.connection.cursor()
        cur.execute("UPDATE images SET consult = %s
            WHERE image_path = %s", ("ANNOTATED",
            request.args.get('path'),))
        mysql.connection.commit()
        cur.close()

    cur = mysql.connection.cursor()
    cur.execute("SELECT * FROM images WHERE consult =
        %s or consult = %s", ("URGENT", "FOR
        ANNOTATION",))
    results = cur.fetchall()
    cur.close()

    return rt('consultation.html', results = results)

class ConsultForm(Form):
    body = TextAreaField('Body', [
        validators.DataRequired(),
        validators.Length(min=10)
        ])
    label = SelectField(u'Label',
        choices = [ (0, 'Without Cardiomegaly'), (1, 'With
            Cardiomegaly')]
        )


@app.route('/consult', methods=['GET', 'POST'])
@is_logged_in
def consult():
    form = ConsultForm(request.form)
    cur = mysql.connection.cursor()
    cur.execute("SELECT image_id from images WHERE
        image_path=%s", (session['temp'],))
    image_id = cur.fetchall()
    cur.close()

    if request.method == 'POST':
        body = form.body.data

        cur = mysql.connection.cursor()
        cur.execute("INSERT INTO messages(image_id, msg,
            from_id) values(%s, %s, %s)", (image_id[0]['
            image_id'], body, session['user_id'],))
        mysql.connection.commit()
        cur.execute("UPDATE images SET consult = %s
            WHERE image_id=%s", ("URGENT", image_id
            [0]['image_id'],))
        mysql.connection.commit()
        cur.close()

        flash("The message has been sent. Wait for the doctor's
            reply", 'success')
        return redirect(url_for('upload'))

    return rt('consult.html', form=form)

@app.route('/annotate', methods=['GET', 'POST'])
@is_logged_in
def annotate():
    form = ConsultForm(request.form)
    img = request.args.get('path')
    reply = request.args.get('reply')

    cur = mysql.connection.cursor()
    cur.execute("SELECT image_id, user_id from images
        WHERE image_path=%s", (img,))
    image_info = cur.fetchone()
    message=''

    for msg in session['messages']:
        if msg['image_id'] == image_info['image_id'] and msg['
            to_id'] == 0:
            message = msg['msg']

    if request.method == 'POST':
```

```
        if reply == 'true':
            cur = mysql.connection.cursor()
            cur.execute("INSERT INTO messages(image_id, msg
                , from_id, to_id) values(%s, %s, %s, %s)", (
                image_info['image_id'], form.body.data, session
                ['user_id'], image_info['user_id']) )
            mysql.connection.commit()
            cur.execute("UPDATE messages SET replied=%s
                WHERE image_id=%s and to_id=%s", ('Y',
                image_info['image_id'],0,))

        cur.execute("UPDATE images SET consult = %s, result
            = %s WHERE image_id=%s", ("ANNOTATED",
            int(form.label.data),image_info['image_id'], ))
        mysql.connection.commit()
        cur.close()

        flash("The image's annotation has been updated", "
            success")
        return redirect(url_for('consultation'))
    return rt('annotate.html', form=form, img=img, reply=
        reply, message=message)


@app.route('/messages')
@is_logged_in
def messages():
    cur = mysql.connection.cursor()
    names = {}
    dates = {}
    read = request.args.get('read')
    msg_id = request.args.get('msg_id')
    cur.execute("SELECT * FROM messages WHERE to_id=%s
        ", ( session['user_id'],))
    session['messages']=cur.fetchall()

    if read:
        cur.execute('SELECT * FROM messages where msg_id
            =%s', (msg_id,))
        message = cur.fetchone()
        cur.execute('SELECT image_path FROM images where
            image_id = %s', (message['image_id'],))
        image_path = cur.fetchone()['image_path']
        cur.execute('UPDATE messages SET replied = %s
            WHERE msg_id=%s', ('Y', msg_id))
        mysql.connection.commit()

        return rt('message.html',message = message['msg'], read
            =True, pid=message['image_id'], image_path=
            image_path)
    else:
        for m in session['messages']:
            cur.execute("SELECT name FROM users WHERE
                user_id = %s", (m['from_id'], ))
            names[m['from_id']] = cur.fetchone()['name']

            cur.execute('SELECT register_date FROM images
                WHERE image_id=%s', (m['image_id'],))
            dates[m['image_id']] = datetime.datetime.strftime(
                cur.fetchone()['register_date'], '%b %d')

        return rt('message.html', messages = session['messages
            '], names = names, dates=dates, read=False)

@app.route('/dataset')
@is_logged_in
def dataset():
    cur = mysql.connection.cursor()
    cur.execute("SELECT * from images where consult = %s",
        ("ANNOTATED",))
    #data = cur.fetchmany(size = 5)
    data = cur.fetchall()

    process = request.args.get('process')

    if process == "make_train_val":
        utilities.make_train_val(data)
        flash("Train and val folders and labels have been
            successfully created", "success")
        cur = mysql.connection.cursor()
        for d in data:
            cur.execute("UPDATE images SET consult=%s
                WHERE image_id=%s", ("FOR TRAINING",
                d['image_id'],))
            mysql.connection.commit()
        cur.close()
        return rt('dataset_2.html')

    elif process == "transfer":
        utilities.transfer_image()
        flash("Annotated images have been transferred
            successfully. " + str(len(data))+ " images has
                been added to the database", 'success')
        return rt('dataset_3.html')

    elif process == "make_lmdb":
        utilities.create_lmdb()
        utilities.convert_binaryproto_to_npy()
        flash("LMDB for Train and Val images have been
            successfully created", 'success')
        return redirect(url_for('dataset'))

    return rt('dataset.html', results = data)

@app.route('/train', methods=['GET', 'POST'])
@is_logged_in
def train():
    json_config = utilities.read_config_file()

    if request.method == 'POST':

        if request.files:
            json_config = utilities.read_config_file()
            files = request.files.getlist('model')
            solver = secure_filename(files[0].filename)
            train_val = secure_filename(files[1].filename)
            utilities.update_model(solver, train_val)

            for f in files:
                model.save(f)

            flash("The system configuration has been updated",
                'success')
            return rt('train.html', config = json_config)

        else:
            flash("No files selected","danger")
            return rt('train.html', config = json_config)

    elif request.args.get('begin'):
            print("here")
            utilities.start_training(json_config['solver'],
                json_config['log'], json_config['parse'])

            lash('Training a new model has been successfully
                executed', 'success')
            return rt('train.html', config = json_config)

    return rt('train.html', config=json_config)

@app.route('/update_config', methods=['GET', 'POST'])
@is_logged_in
def update_config():
    json_config = utilities.read_config_file()

    if request.method == 'POST':
        json_config = utilities.read_config_file()
        if request.files:

            files = request.files.getlist('model')
            deploy = secure_filename(files[0].filename)
            weights = secure_filename(files[1].filename)
            utilities.update_config_file(deploy, weights)
            json_config = utilities.read_config_file()

            for f in files:
                print(f)
                model.save(f)
            flash('The system configuration has been updated',
                'success')
            return rt('update_config.html', config=json_config)

        else:

            flash("No files selected","danger")
            return rt('update_config.html', config=json_config)

    return rt('update_config.html', config=json_config)

@app.route('/batch')
@is_logged_in
def batch():
    stat = utilities.compute_accuracy()
    if request.args.get('begin'):
        utilities.start_batch_predict()
        stat = utilities.compute_accuracy()
        # [accuracy, sensitivity, specificity]
        return rt('batch.html', stat=stat, new = True)
    return rt('batch.html', stat=stat, new = False)


@app.route('/logout')
def logout():
    session.clear();
```

```
    return redirect(url_for('index'))

if __name__ == '__main__':
    app.secret_key='hCoK4rZbRsRGgZfd'
    app.run()
```

utilities.py

```
import os
import json
import csv

from math import ceil
from random import shuffle
from shutil import copy, move
from operator import itemgetter
from subprocess import call, Popen

def make_train_val(data):

    os.system('mkdir "./static/img/train"')
    os.system('mkdir "./static/img/val"')

    f = open('static/img/train.txt', 'w')
    e = open('static/img/val.txt', "w")

    trainLen = ceil(len(data)*0.6)

    train = data[0:trainLen]
    val = data[trainLen:]

    for t in train:
        f.write("\n"+t['image_path'].split('img/')[1] + " " +
              str(t['result']))
        copy("./"+t['image_path'], './static/img/train')

    for v in val:
        e.write("\n"+v['image_path'].split('img/')[1] + " " +
              str(v['result']))
        copy("./"+v['image_path'], './static/img/val')

    f.close()
    e.close()

def transfer_image():

    for t in os.listdir("./static/img/train"):
        move("./static/img/train/"+t, "./caffe/data/
              heart_enlargement/train")

    for v in os.listdir("./static/img/val"):
        move("./static/img/val/"+v, "./caffe/data/
              heart_enlargement/val")

    trainNew = open("./static/img/train.txt", "r")
    valNew = open("./static/img/val.txt", "r")

    trainOld = open("./caffe/data/heart_enlargement/train.txt",
          "a+")
    valOld = open("./caffe/data/heart_enlargement/val.txt", "a
         +")

    for t in trainNew:
        trainOld.write(t)

    for v in valNew:
        valOld.write(v)

    trainOld.close()
    trainNew.close()
    valOld.close()
    valNew.close()

def create_lmdb():
    call(['git-bash','create_lmdb.sh'])

def convert_binaryproto_to_npy():
    CONVERT = "./caffe/python/convert.py"
    PROTO = "./caffe/data/heart_enlargement/heart_mean.
          binaryproto"
    NPY = "./caffe/data/heart_enlargement/heart_mean.npy"
    call(['python', CONVERT, PROTO, NPY])

def read_config_file():
    with open('config.json') as d:
        config = json.load(d)

    return config

def update_config_file(deploy, weights):
    with open('config.json', 'r+') as conf:
        config = json.load(conf)
        wei = config['weights']
        w = list(wei.split('heart_enlargement_model/'))

        dep = config['deploy']
        d = list(dep.split('heart_enlargement_model/'))

        w[1] = "heart_enlargement_model/" + weights
        config['weights'] = ''.join(w)

        d[1] = "heart_enlargement_model/" + deploy
        config['deploy'] = ''.join(d)

        conf.seek(0)
        json.dump(config, conf)
        conf.truncate()

def update_model(solver, train_val):
    with open('config.json', 'r+') as conf:
        config = json.load(conf)
        old_solver = config['solver']
        s = list(old_solver.split('heart_enlargement_model/'))

        old_tv = config['train_val']
        tv = list(old_solver.split('heart_enlargement_model/'))

        s[1] = "heart_enlargement_model/" + solver
        config['solver'] = ''.join(s)

        tv[1] = "heart_enlargement_model/" + train_val
        config['train_val'] = ''.join(tv)

        conf.seek(0)
        json.dump(config, conf)
        conf.truncate()

def start_training(solver, log, parse):
    Popen(['git-bash', 'train.sh', solver, log])

def start_batch_predict():
    predict = "./caffe/python/predict.py"

    config = read_config_file()

    call(['python', predict, config['binary_mean'], config['
          deploy'], config['weights']])

def compute_accuracy():
    test = list(csv.reader(open('test.csv', 'r')))
    predict = list(csv.reader(open('predict.csv', 'r')))
    a =[]
    b=[]

    for t in test:
        a.append(list(map(int, t)))
    a = sorted(a, key=lambda a: a[0])

    for p in predict[1:]:
        b.append(list(map(int, p)))
    b = sorted(b, key=lambda b:b[0])

    trueP = 0
    trueN = 0
    falseP = 0
    falseN = 0

    for t, p in zip(a, b):
        if t[0] == p[0]:
            if t[1] == p[1]:
                print("True value: ",t, p)
                if p[1] == 0:
                    trueN += 1
                else:
                    trueP += 1
            else:
                print("False value: ", t, p)
                if p[1] == 0:
                    falseN += 1
                else:
                    falseP += 1
    accuracy = (trueP+trueN) / (trueN+trueP+falseN+falseP)
    sensitivity = trueP / (trueP+falseN)
    specificity = trueN / (trueN+falseP)

    return [accuracy, sensitivity, specificity]
```

create_lmdb.sh

```
#!/usr/bin/env sh
# Create the imagenet lmdb inputs
# N.B. set the path to the imagenet train + val data dirs

CAFFE_ROOT=./caffe
```

```
echo "${PWD}"

DATA=$CAFFE_ROOT/data/heart_enlargement/
TOOLS=$CAFFE_ROOT/scripts/build/tools/Release

TRAIN_DATA_ROOT=$DATA/train/
VAL_DATA_ROOT=$DATA/val/

# Set RESIZE=true to resize the images to 256x256. Leave as
      false if images have
# already been resized using another tool.
RESIZE=true
if $RESIZE; then
  RESIZE_HEIGHT=224
  RESIZE_WIDTH=224
else
  RESIZE_HEIGHT=0
  RESIZE_WIDTH=0
fi

if [ ! -d "$TRAIN_DATA_ROOT" ]; then
  echo "Error: TRAIN_DATA_ROOT is not a path to a directory
       : $TRAIN_DATA_ROOT"
  echo "Set the TRAIN_DATA_ROOT variable in
       create_imagenet.sh to the path" \
       "where the ImageNet training data is stored."
  exit 1
fi

if [ ! -d "$VAL_DATA_ROOT" ]; then
  echo "Error: VAL_DATA_ROOT is not a path to a directory:
       $VAL_DATA_ROOT"
  echo "Set the VAL_DATA_ROOT variable in create_imagenet.
       sh to the path" \
       "where the ImageNet validation data is stored."
  exit 1
fi

echo "Creating train lmdb..."

rm -fr $DATA/train_lmdb

GLOG_logtostderr=1 $TOOLS/convert_imageset \
    --resize_height=$RESIZE_HEIGHT \
    --resize_width=$RESIZE_WIDTH \
    --shuffle \
    $TRAIN_DATA_ROOT \
    $DATA/train.txt \
    $DATA/train_lmdb

echo "Creating val lmdb..."

rm -fr $DATA/val_lmdb

GLOG_logtostderr=1 $TOOLS/convert_imageset \
    --resize_height=$RESIZE_HEIGHT \
    --resize_width=$RESIZE_WIDTH \
    --shuffle \
    $VAL_DATA_ROOT \
    $DATA/val.txt \
    $DATA/val_lmdb


$TOOLS/compute_image_mean $DATA/train_lmdb \
  $DATA/heart.binaryproto

echo "Done."
```

config.json

```
{
    "result": "./static/result.npy",
    "classify": "./caffe/python/classify.py",
    "parse": "./caffe/tools/extra/parse_log.py",
    "caffe": "caffe/scripts/build/tools/Release/caffe",
    "mean": "./caffe/data/heart_enlargement/heart_mean.npy",
    "log": "./caffe/models/heart_enlargement_model/solver_new
          .log",
    "deploy": "./caffe/models/heart_enlargement_model/deploy.
          prototxt",
    "solver": "./caffe/models/heart_enlargement_model/solver.
          prototxt",
    "train_val": "./caffe/models/heart_enlargement_model/
          train_val.prototxt",
    "binary_mean": "./caffe/data/heart_enlargement/heart_mean.
          binaryproto",
    "weights": "./caffe/models/heart_enlargement_model/
          new_model_iter_4000.caffemodel"
}
```

templates/annotate.html

```
{% extends 'logged-layout.html' %} {% block body %}
{% if reply == 'true' %}
<h1>Reply to a Patient</h1>
{% else %}
<h1>Annotate Image</h1>
{% endif %}
{% from 'includes/_formhelpers.html' import render_field %}
<div class="row">
    <div class="col">
        <img src="{{img}}" style="height:400px; margin-left
            :-15px;" class="rounded border border-primary
            "/>
    </div>

    <div class="col">
        <form class="form-signin" method="POST" action
            "">
            <div class="form-group">
                <label for="label">Label:</label>
                {{render_field(form.label, class_="form-
                    control", id="label")}}
            </div>
            {% if reply == 'true' %}
                <div class="form-group">
                    <label for="disabledTextInput">Patient's
                        Message:</label>
                    <textarea class="form-control" id="
                        exampleFormControlTextarea1" rows
                        ="3" disabled>{{message}}</
                        textarea>
                </div>

                <div class="form-group">
                    <label for="editor">Reply:</label>
                    {{render_field(form.body, class_="form-
                        control", id="editor")}}
                </div>
            {% endif %}
            <p>
                <input type="submit" class="btn btn-outline
                    -primary" value="Submit">
            </p>
        </form>
    </div>
</div>

{% endblock %}
```

templates/batch.html

```
{% extends 'logged-layout.html' %}
{% block body %}
<h2 class="display-4 text-primary">Start Batch Prediction</
    h2>
<p class="lead text-secondary">
    This is to test the model's <i>Accuracy</i>, <i>
        Sensitivity</i>, and <i>Specificity</i> if given new
        inputs.
</p>
<hr>
    <div class='row' style="margin-top: 5%">
        <div class="col text-center">
            <a href="{{url_for('batch')}}?begin=true" class="
                btn btn-outline-success">Start Batch
                Prediction</a>
        </div>
        <div class="col" style="margin-right: 10%">
            {% if new %}
                <div class="card">
                    <h5 class="card-header text-primary text
                        -center">Previous Model's Statistics
                        </h5>
                    <div class="card-body">
                        <p class="card-text lead text-
                            secondary">
                            Accuracy: {{statOld[0] * 100}} %
                            <br>Sensitivity: {{statOld[1] *
                                100}} %
                            <br>Specificity: {{statOld[2] *
                                100}} %

                        </p>
                    </div>
                </div>
                <div class="card">
                    <h5 class="card-header text-primary text
                        -center">New Model's Statistics</h5
                        >
                    <div class="card-body">
                        <p class="card-text lead text-
                            secondary">
                            Accuracy: {{statNew[0] * 100}} %
```

```
                        <br>Sensitivity: {{statNew[1] *
                            100}} %
                        <br>Specificity: {{statNew[2] *
                            100}} %

                    </p>
                </div>
            </div>
        {% else %}
            <div class="card">
                <h5 class="card-header text-primary text
                    -center">Current Model's Statistics
                    </h5>
                <div class="card-body">
                    <p class="card-text lead text-
                        secondary">
                        Accuracy:  {{statOld[0] * 100}} %
                        <br>Sensitivity: {{statOld[1] *
                            100}} %
                        <br>Specificity: {{statOld[2] *
                            100}} %

                    </p>
                </div>
            </div>
        {% endif %}
    </div>

{% endblock %}
```

templates/consult.html

```
{% extends 'logged-layout.html' %}
{% block body %}
    <h2 class="display-4 text-primary">Consult to a doctor
        </h2>
    <p class="lead text-secondary">
        Compose a message regarding your concern about the
            system-provided result
    </p>
    <hr>

    {% from 'includes/_formhelpers.html' import render_field %}
    <div class="row">
        <div class="col">
            <img src="{{session['temp']}}" style="height:400px
                ; margin-left:-15px;" class="rounded border
                border-primary"/>
        </div>

        <div class="col">
            <form class="form-signin" method="POST" action
                ="">
                <div class="form-group">
                    <label for="editor">Your Message to the
                        Doctor:</label>
                    {{render_field(form.body, class_="form-
                        control", id="editor")}}
                </div>
                <p>
                    <input type="submit" class="btn btn-
                        outline-primary" value="Submit">
                </p>
            </form>
        </div>
    </div>

{% endblock %}
```

templates/consultation.html

```
{% extends 'logged-layout.html' %}
{% block body %}
<style>
.card-img-top {
    width: 100%;
    height: 15vw;
    object-fit: cover;
}
</style>
<h2 class="display-4 text-primary">Consultations</h2>
<p class="lead text-secondary">
    Below shows the images uploaded to and annotated by the
        system.The images are subclassified into two status.
    <br><strong class="text-primary">FOR ANNOTATION
        </strong> status are images annotated by the system
        accepted by the client outright.
    <br> With this, you can either accept the prediction
        provided or change its annotation.
    <br><strong class="text-danger">URGENT</strong>
        status are images with message from the client
        regarding their concern with the system-provided
        result.
</p>
<hr>
<div class="row">
    {% for result in results[::-1] %}

    <div class="mx-auto mt-3">
        {% if result['consult'] == 'URGENT' %}
        <div class="card border-danger" style="width: 18rem
            ;">
            <img class="card-img-top" src={{result['
                image_path']}} alt="Card image cap">
            <div class="card-body text-danger">
                <h5 class="card-title">X-Ray Summary</h5
                    >
                <p class="card-text">

                    {% if result['result'] == 0 %} Result:
                    <strong> Without Cardiomegaly </strong>
                    {% else %} Result:
                    <strong>With Cardiomegaly </strong>
                    {% endif %}
                    <br> Status:
                    <strong>{{result['consult']}} </strong>
                </p>
                <div class="card-footer text-center">
                    <a href="{{url_for('annotate')}}?path={{
                        result['image_path']}}&amp;reply=
                        true" class="btn btn-outline-danger
                        ">Reply</a>

                </div>
        {% else %}
        <div class="card border-primary" style="width: 18rem
            ;">
            <img class="card-img-top" src={{result['
                image_path']}} alt="Card image cap">
            <div class="card-body text-primary">
                <h5 class="card-title">X-Ray Summary</h5
                    >
                <p class="card-text">

                    {% if result['result'] == 0 %} Result:
                    <strong> Without Cardiomegaly </strong>
                    {% else %} Result:
                    <strong>With Cardiomegaly </strong>
                    {% endif %}
                    <br> Status:
                    <strong>{{result['consult']}} </strong>
                </p>
                <div class="card-footer text-center">
                    <a href="{{url_for('consultation')}}?path
                        ={{result['image_path']}}" class="btn
                         btn-outline-success">Accept</a>
                    <a href="{{url_for('annotate')}}?path={{
                        result['image_path']}}&amp;reply=
                        false" class="btn btn-outline-danger
                        ">Annotate</a>
                </div>
        {% endif %}

            </div>
        </div>
    </div>
    {% endfor %}
</div>
{% endblock %}
```

templates/dashboard.html

```
{% extends 'logged-layout.html' %}

{% block body %}
    <h1 class="display-1 text-primary">Standard User</h1>
    <hr>
    <p class="lead">
        As a standard user, you are granted access to following
            functionalities :
        <br>
        <a href="/upload">X-Ray Prediction</a><br>
        <a href="/messages">Messages</a><br>
    </p>
    <div class="navbar navbar-fixed-bottom">
        <p class="mt-5 mb-3 text-muted" style="padding-
            top: 20%">
            For further questions or inquiries, you may send
                an email to <a href="mailto:elacanlale@up.
                edu.ph">elacanlale@up.edu.ph</a>
        </p>
    </div>


{% endblock %}
```

templates/dashboard-admin.html

```
{% extends 'logged-layout.html' %}
{% block body %}
<h1 class="display-1 text-primary">AI Expert</h1>
<hr>
<p class="lead">
    As an AI Expert, you are granted access to following
        functionalities :
    <br>
    <a href="/upload">X-Ray Prediction</a>
    <br>
    <a href="/dataset">Update Database</a>
    <br>
    <a href="/train">Train the Model</a>
    <br>
    <a href="/update_config">Update the System
        Configuration</a>
    <br>
    <a href="/batch">Batch Prediction</a>
    <br>
</p>
<div class="navbar navbar-fixed-bottom">
    <p class="mt-5 mb-3 text-muted" style="padding-top:
        12%">
        For further questions or inquiries, you may send an
            email to
        <a href="mailto:elacanlale@up.edu.ph">elacanlale@up.
            edu.ph</a>
    </p>
</div>

{% endblock %}
```

templates/dashboard-doctor.html

```
{% extends 'logged-layout.html' %}
{% block body %}
<h1 class="display-1 text-primary">Doctor</h1>
<hr>
<p class="lead">
    As a doctor, you are granted access to following
        functionalities :
    <br>
    <a href="/upload">X-Ray Prediction</a>
    <br>
    <a href="/consultation">Consultation</a>
    <br>
</p>
<div class="navbar navbar-fixed-bottom">
    <p class="mt-5 mb-3 text-muted" style="padding-top:
        20%">
        For further questions or inquiries, you may send an
            email to
        <a href="mailto:elacanlale@up.edu.ph">elacanlale@up.
            edu.ph</a>
    </p>
</div>

{% endblock %}
```

templates/dataset.html

```
{% extends 'logged-layout.html' %}
{% block body %}
<style>
    .card-img-top {
        width: 100%;
        height: 10vw;
        object-fit: cover;
    }
</style>
<div>
    <h2 class="display-4 text-primary">Create Training and
        Validation Set</h2>
    <p class="lead text-secondary">
        These are the images with confirmed labels by a doctor.
            Seperate the annotated dataset into training and
            validation set with its label.
    </p>
    <a href="{{url_for('dataset')}}?process=make_train_val"
        class="btn btn-outline-danger" data-toggle="
        tooltip" title="Seperate the annotated dataset into
        training and validation set with its label">Create
        Training and Validation</a>
</div>
<hr>
<div class="row">
    {% for result in results %}
    <div class="mx-auto mt-5">
        <div class="card border-primary" style="width: 10rem
            ;">
            <img class="card-img-top" src={{result['
                image_path']}} alt="Card image cap">
            <div class="card-body text-primary">
                <h5 class="card-title">X-Ray Summary</h5
                    >
                <p class="card-text">

                    {% if result['result'] == 0 %} Result:
                    <strong> Without Cardiomegaly </strong>
                    {% else %} Result:
                    <strong>With Cardiomegaly </strong>
                    {% endif %}
                </p>
            </div>
        </div>
    </div>
    {% endfor %}
</div>

{% endblock %}
```

templates/dataset_2.html

```
{% extends 'logged-layout.html' %} {% block body %}
<div>
    <h2 class="display-4 text-primary">Transfer the images
        </h2>
    <p class="lead text-secondary">
        Once the images are separated into different sets, we'll
            transer the images and label to Caffe's data
            folder.
    </p>
    <a href="{{url_for('dataset')}}?process=transfer" class="
        btn btn-outline-warning" data-toggle="tooltip"
        title="Append the separated training and validation
        images to caffe data folder">Transfer the images </a
        >
</div>
<hr>
{% endblock %}
```

templates/dataset_3.html

```
{% extends 'logged-layout.html' %} {% block body %}
<div>
    <h2 class="display-4 text-primary">Create LMDB and
        Compute Image Mean</h2>
    <p class="lead text-secondary">
        We need to update the LMDB and Mean file of the
            current dataset. LMDB stands for Lightning-
            Mapped Database.
        LMDB is the database of choice when using Caffe with
            large datasets. <br> Also, the model requires us to
            subtract the image mean from each image, so we
            have to compute the mean
    </p>
    <a href="{{url_for('dataset')}}?process=make_lmdb" class
        ="btn btn-outline-success" data-toggle="tooltip"
        title="Update the train and val LMDB and its mean
        file">Create LMDB and Mean File</a>
</div>
<hr>
{% endblock %}
```

templates/home.html

```
{% extends 'layout.html' %}
{% block body %}
<div>
  <div class="row" style="padding-top: 5%">

    <div class="col" style="padding-top: 5%">
        <h1 class="display-3 text-primary">
          medicine <br>meets <br>technology
        </h1>
        <p class="lead">
          improving the quality of life, one app at a time
        </p>
        {% if session.logged_in == NULL %}
        <a href="/register" class="btn btn-outline-primary">
          Register
          <span class="lnr lnr-arrow-down"></span>
        </a>
        <a href="/login" class="btn btn-outline-primary">
          <span class="lnr lnr-arrow-up"></span>
          Login
        </a>
        {% endif %}
    </div>
    <div class="col">
        <img class="d-block w-100" src="static/logo/staff4.
            png">
    </div>
```

```
  </div>
  <p class="lead text-center text-muted" style="padding-top
      : 5%">&copy; University of the Philippines Manila</p>
</div>

{% endblock %}
```

templates/layout.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>HeartSmart</title>
    <link rel="stylesheet" href="{{url_for('static', filename='
        bootstrap-4.0.0-dist/css/bootstrap.css')}}">

    {% include 'includes/_navbar.html' %}
  </head>
  <body>
    <div class="container">
      <div>
        {% include 'includes/_messages.html' %}
      </div>

      {% block body %}
      {% endblock %}
    </div>

    <script src="{{url_for('static', filename='bootstrap
        -4.0.0-dist/js/jquery-3.2.1.slim.min.js')}}"></
        script>
    <script src="{{url_for('static', filename='bootstrap
        -4.0.0-dist/js/popper.min.js')}}"></script>
    <script src="{{url_for('static', filename='bootstrap
        -4.0.0-dist/js/bootstrap.min.js')}}"></script>
    <script src="{{url_for('static', filename='bootstrap
        -4.0.0-dist/node_modules/feather-icons/dist/feather.
        min.js')}}"></script>
    <script src="{{url_for('static', filename='bootstrap
        -4.0.0-dist/js/alert.js')}}"></script>
    <script>
      feather.replace()
    </script>

  </body>
</html>
```

templates/logged-layout.html

```
<!doctype html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,
        initial-scale=1, shrink-to-fit=no">
    <meta name="description" content="">
    <meta name="author" content="">

    <title>HeartSmart</title>

    <!-- Bootstrap core CSS -->
    <link rel="stylesheet" href="{{url_for('static', filename='
        bootstrap-4.0.0-dist/css/bootstrap.css')}}">
    <link rel="stylesheet" href="{{url_for('static', filename='
        bootstrap-4.0.0-dist/css/dashboard.css')}}">
</head>

<body>
<nav class="navbar navbar-dark sticky-top bg-dark flex-md
    -nowrap p-0">
    <a class="navbar-brand col-sm-3 col-md-2 mr-0" href
        ="/">
        <span data-feather="activity"></span> HeartSmart
            </a>
    <ul class="navbar-nav col col-lg-1" >
        <li class="nav-item">
            <a class="nav-link" href="/logout">
                <span data-feather="log-out"></span>
                Sign out
            </a>
        </li>
    </ul>
</nav>
<div class="container-fluid">
    <div class="row">
        <nav class="col-md-2 d-none d-md-block bg-light
            sidebar">
            <div class="sidebar-sticky">
                <ul class="nav flex-column">
                    <li class="nav-item">
                        <a class="nav-link active" href="/
                            dashboard">
                            <span data-feather="home"></
                                span>
                            {{session['name']}}'s Dashboard
                            <span class="sr-only">(current)
                                </span>
                        </a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="/upload">
                            <span data-feather="file"></span
                                >
                            X-Ray Prediction
                        </a>
                    </li>

                    <!--<li class="nav-item">
                        <a class="nav-link" href="/summary
                            ">
                            <span data-feather="bar-chart
                                -2"></span>
                            Summary
                        </a>
                    </li>-->
                    {% if session['user_type'] == 'admin' %}
                        <h6 class="sidebar-heading d-flex
                            justify-content-between align-
                            items-center px-3 mt-4 mb-1
                            text-muted">
                            <span>Train Network</span>
                            <a class="d-flex align-items-
                                center text-muted" href
                                ="#">
                            </a>
                        </h6>
                        <ul class="nav flex-column mb-2">
                            <li class="nav-item">
                                <a class="nav-link" href="/
                                    dataset">
                                    <span data-feather="
                                        database"></span>
                                    Update Dataset
                                </a>
                            </li>
                            <li class="nav-item">
                                <a class="nav-link" href="/
                                    train">
                                    <span data-feather="cpu
                                        "></span>
                                    Train the Model
                                </a>
                            </li>
                            <li class="nav-item">
                                <a class="nav-link" href="/
                                    update_config">
                                    <span data-feather="hard
                                        -drive"></span>
                                    Update the System Config
                                </a>
                            </li>
                            <li class="nav-item">
                                <a class="nav-link" href="/
                                    batch">
                                    <span data-feather="
                                        server"></span>
                                    Batch Prediction
                                </a>
                            </li>
                        </ul>
                    {% elif session['user_type'] == 'doctor'%}
                        <li class="nav-item">
                            <a class="nav-link" href="/
                                consultation">
                                <span data-feather="monitor
                                    "></span>
                                Consultation
                                <span class="badge badge-
                                    success badge-pill" data
                                    -toggle="tooltip" title="
                                    Normal Consultation">{{
                                    session['consult'] | length
                                    }}</span>
                                <span class="badge badge-
                                    danger badge-pill" data-
                                    toggle="tooltip" title="
                                    Urgent Consultation">{{
                                    session['urgent'] | length
                                    }}</span>
                            </a>
                        </li>
                    {% else %}
```

```
                    <li class="nav-item">
                        <a class="nav-link" href="/messages
                            ">
                            <span data-feather="mail"></
                                span>
                            Messages
                            <span class="badge badge-primary
                                badge-pill" data-toggle="
                                tooltip" title="New Messages
                                ">{{session['messages'] |
                                length}}</span>
                        </a>
                    </li>
                    {% endif %}

                </ul>
            </div>
        </nav>

        <main role="main" class="col-md-9 ml-sm-auto col
            -lg-10 pt-3 px-4">
            <div class="d-flex justify-content-between flex-
                wrap flex-md-nowrap align-items-center pb
                -2 mb-3 border-bottom">
                <!--<h1 class="h2">Dashboard</h1>
                <div class="btn-toolbar mb-2 mb-md-0">
                    <div class="btn-group mr-2">
                        <button class="btn btn-sm btn-
                            outline-secondary">Share</
                            button>
                        <button class="btn btn-sm btn-
                            outline-secondary">Export</
                            button>
                    </div>
                    <button class="btn btn-sm btn-outline-
                        secondary dropdown-toggle">
                        <span data-feather="calendar"></
                            span>
                        This week
                    </button>
                </div>-->
            </div>
            {% include 'includes/_messages.html' %}
            {% block body %}
            {% endblock %}

        </main>
    </div>
</div>


    <!-- Bootstrap core JavaScript
    ================================================== -->
    <!-- Placed at the end of the document so the pages load
        faster -->
    <script src="{{url_for('static', filename='bootstrap
        -4.0.0-dist/js/jquery-3.2.1.slim.min.js')}}"></
        script>
    <script src="{{url_for('static', filename='bootstrap
        -4.0.0-dist/js/popper.min.js')}}"></script>
    <script src="{{url_for('static', filename='bootstrap
        -4.0.0-dist/js/bootstrap.min.js')}}"></script>
    <script src="{{url_for('static', filename='bootstrap
        -4.0.0-dist/node_modules/feather-icons/dist/feather.
        min.js')}}"></script>
    <script src="{{url_for('static', filename='bootstrap
        -4.0.0-dist/js/alert.js')}}"></script>
    <script>
        feather.replace()
    </script>
    <!--<script src="{{url_for('static', filename='ckeditor/
        ckeditor.js')}}"></script>
    <script type="text/JavaScript">
        CKEDITOR.replace('editor')
    </script>-->

</body>

</html>
```

templates/login.html

```
<link rel="stylesheet" href="{{url_for('static', filename='
    bootstrap-4.0.0-dist/css/signin.css')}}">

{% extends 'layout.html' %}

{% block body %}

<div class="text-center" style="margin-left: 350px; margin-
    right: 350px">
    <!--<img src="{{url_for('static', filename='logo/upm.png')
        }}" />-->
    <form action="" class="form-signin" method="POST">
    <img class="mb-4" src="{{url_for('static', filename='logo/
        upm.png')}} " width="200" height="200"style="margin
        -top:10%; margin-bottom: 5%" alt="">

    <div class="form-group">
        <input type="text" name='username' class="form-
            control" placeholder="Username" value={{request.
            form.username}} >
      </div>
      <div class="form-group">
        <input type="password" name = 'password' class="form
            -control" placeholder="Password" value={{request.
            form.password}}>
      </div>
      <button type="submit" class="btn btn-lg btn-outline-
          primary btn-block">Login</button>
      <p class="mt-5 mb-3 text-muted">&copy; University of
          the Philippines Manila</p>
    </form>
</div>
{% endblock %}
```

templates/message.html

```
{% extends 'logged-layout.html' %}
{% block body %}

    {% if read %}
        <div class="row">
            <div class="col">
                <div class="card border-success" style="width
                    : 25rem;">
                    <img class="card-img-top" src={{
                        image_path}} alt="Card image cap">
                    <div class="card-body text-success">
                        <h5 class="card-title">Patient ID: {{
                            pid}}</h5>
                    </div>
                </div>
            </div>

            <div class="col" style="margin-top: 10%; margin-
                right: 5%">

                <div class="form-group">
                    <label for="disabledTextInput">Doctor's
                        Message:</label>
                    <textarea class="form-control" id="
                        exampleFormControlTextarea1" rows
                        ="3" disabled>{{message}}</
                        textarea>
                    <a href="{{url_for('messages')}}" class="
                        btn btn-outline-danger" style="
                        margin-top: 5%">Dismiss</a>
                </div>


            </div>
        </div>
    {% else %}
    <h2 class="display-4 text-primary">Messages</h2>
    <p class="lead text-secondary">
        These are the replies from the doctor regarding your
            concerns with the result provided by the system
    </p>
    <hr>
        {% for message in messages[::-1] %}

            <div class="my-3 p-3 bg-white rounded box-
                shadow">
                <div class="media text-muted pt-3">
                    <p class="media-body shadow pb-3 mb-0
                        border-bottom border-gray">
                        <a href="{{url_for('messages')}}?read=
                            true&amp;msg_id={{message['
                            msg_id']}}">
                            {% if message['replied'] == 'N' %}
                                <strong class="d-block text-
                                    primary">
                                    <span data-feather="user
                                        "></span>
                                    {{names[message['from_id
                                        ']]}} <br>
                                    Re: [ Patient {{message['
                                        image_id']}} ]
                                </strong>
                            {% else %}
                                <strong class="d-block text-
                                    secondary">
```

```
                                        <span data-feather="user
                                            "></span>
                                        {{names[message['from_id
                                            ']]}} <br>
                                         Re: [ Patient {{message['
                                            image_id']}} ]
                                    </strong>
                                {% endif %}

                            </a>
                            {{message['msg']}}
                            <div class="text-right">{{dates[
                                message['image_id']]}}</div>
                        </p>
                    </div>
                </div>
            {% endfor %}
        {% endif %}
{% endblock %}
```

templates/register.html

```
<link rel="stylesheet" href="{{url_for('static', filename='
    bootstrap-4.0.0-dist/css/starter-template.css')}}">
 {% extends 'layout.html' %}

{% block body %}
  <h1 class="display-4 text-primary text-center">Register</
      h1>
  {% from 'includes/_formhelpers.html' import render_field %}
  <form class="form-signin" method="POST" style="padding
      -left: 300px; padding-right: 300px">
    <div class="form-group">
      <label for="inputName">Name:</label>
      {{render_field(form.name, class_="form-control", id="
          inputName")}}
    </div>
    <div class="form-group">
      <label for="inputEmail">Email:</label>
      {{render_field(form.email, class_="form-control", id="
          inputEmail")}}
    </div>
    <div class="form-group">
      <label for="inputUname">Username:</label>
      {{render_field(form.username, class_="form-control", id
          ="inputUname")}}
    </div>
    <div class="form-group">
      <label for="inputUtype">User Type:</label>
      {{render_field(form.user_type, class_="form-control", id
          ="inputUtype")}}
    </div>
    <div class="form-inline"></div>
    <div class="row">
      <div class="col">
        <label for="inputPassword">Password:</label>
        {{render_field(form.password, class_="form-control",
            id="inputPassword")}}
      </div>
      <div class="col">
        <label for="confirmPass">Confirm Password:</label>
        {{render_field(form.confirm, class_="form-control", id
            ="confirmPass")}}
      </div>
    </div>

    <p class="text-center" style="padding-top: 3%">
      <input type="submit" class="btn btn-outline-primary"
          value="Submit">
    </p>
    <p class="text-center text-muted">&copy; University of
        the Philippines Manila</p>
  </form>
{% endblock %}
```

templates/summary.html

```
{% extends 'logged-layout.html' %}
{% block body %}
<div class="row">
    {% for path in img_paths %}

        <div class="mx-auto">
            <div class="card" style="width: 18rem;">
                <img class="card-img-top" src={{path['
                    image_path']}} alt="Card image cap">
                <div class="card-body">
                    <p class="card-text">Some quick example
                        text to build on the card title and
                        make up the bulk of the card's content
                        .</p>
                </div>
            </div>
        </div>
    {% endfor %}
</div>
{% endblock %}
```

templates/train.html

```
{% extends 'logged-layout.html' %}
{% block body %}
<h2 class="display-4 text-primary">Train the Model</h2>
<p class="lead text-secondary">
  Before starting training for a new model, it requires a <i>
      Solver prototxt</i>
  file and a <i>Train-Val prototxt</i> file<br>
  Solver prototxt contains the parameters for training.
  (For more information about the parameters, click
  <a href="" data-toggle="modal" data-target="#
      exampleModal"> here </a> )<br>
  Train-Val prototxt describes the network architecture to be
      used during training phase.

</p>
<hr>
    <div class='row'>
        <div class="col">
            <label>Solver file:</label>
            <input class="form-control" type="text"
                placeholder={{config['solver']}} readonly>
            <br>
            <label>Train-Val file:</label>
            <input class="form-control" type="text"
                placeholder={{config['train_val']}} readonly>
            <br>
        </div>
        <div class="col" style="margin-top: 25px">
            <form action="{{url_for('train')}}" method="post"
                enctype="multipart/form-data">
                <div class="form-group">
                    <label for="solver">
                        <strong> Choose new solver prototxt
                            :</strong>
                    </label>
                    <br>
                    <input type="file" name="model" accept
                        =".prototxt" id="solver" class="form
                        -control-file">
                </div>
                <div class="form-group">
                    <label for="train_val">
                        <strong> Choose new train_val prototxt
                            :</strong>
                    </label>
                    <br>
                    <input type="file" name="model" accept
                        =".prototxt" id="train_val" class="
                        form-control-file">
                </div>
                <input type="submit" value="Update" class="
                    btn btn-outline-primary">
                <a href="{{url_for('train')}}?begin=true" class
                    ="btn btn-outline-success">Start
                    Training</a>
            </form>
        </div>
    </div>
    <!-- Modal -->
    <div class="modal fade" id="exampleModal" tabindex
        ="-1" role="dialog" aria-labelledby="
        exampleModalLabel" aria-hidden="true">
        <div class="modal-dialog modal-lg" role="document
            ">
            <div class="modal-content">
                <div class="modal-header">
                    <h3 class="lead text-primary" id="
                        exampleModalLabel"><strong>
                        Solver Parameters: </strong></h3>
                    <button type="button" class="close" data
                        -dismiss="modal" aria-label="Close
                        ">
                        <span aria-hidden="true">&times;</
                            span>
                    </button>
                </div>
                <div class="modal-body lead text-secondary
                    ">
                    <p>
                        <dl class="row">
                            <dt class="col-sm-3">base_lr</
                                dt>
                            <dd class="col-sm-9">
                                This parameter indicates the
                                    base (beginning) learning
                                    rate of the network.
```

```
                    The value is a real number (
                        floating point).
                </dd>

                <dt class="col-sm-3">lr_policy</
                    dt>
                <dd class="col-sm-9">
                    This parameter indicates how
                        the learning rate should
                        change over time.
                    This value is a quoted string.
                </dd>

                <dt class="col-sm-3">stepsize</
                    dt>
                <dd class="col-sm-9">
                    This parameter indicates how
                        often (at some iteration
                        count)
                    that we should move onto the
                        next "step" of training.
                    This value is a positive
                        integer.
                </dd>

                <dt class="col-sm-3">stepvalue
                    </dt>
                <dd class="col-sm-9">
                    This parameter indicates one of
                         potentially many
                        iteration counts
                    that we should move onto the
                        next "step" of training.
                    This value is a positive integer
                        .
                    There are often more than one
                        of these parameters
                        present,
                    each one indicated the next step
                        iteration.
                </dd>

                <dt class="col-sm-3">max_iter</
                    dt>
                <dd class="col-sm-9">
                    This parameter indicates when
                        the network should stop
                        training.
                    The value is an integer
                        indicate which iteration
                        should
                    be the last.
                </dd>

                <dt class="col-sm-3">snapshot
                    </dt>
                <dd class="col-sm-9">
                    This parameter indicates how
                        often caffe should output
                         a model and solverstate.
                    This value is a positive
                        integer.
                </dd>

                <dt class="col-sm-3">
                    snapshot_prefix</dt>
                <dd class="col-sm-9">
                    This parameter indicates how a
                         snapshot output's model
                        and
                    solverstate's name should be
                        prefixed. This value is a
                        double
                    quoted string.
                </dd>
            </dl>
        </p>
    </div>
</div>
</div>
</div>

{% endblock %}
```

templates/update_config.html

```
{% extends 'logged-layout.html' %} {% block body %}
<h2 class="display-4 text-primary">Update the System
    Configuration</h2>
<p class="lead text-secondary">
    After training a new model, you can update the model used
        and provide its corresponding deploy prototxt.
    <br> This will be saved as the new new model to be used
        by the system.
</p>
<hr>

<div class='row'>
    <div class="col">
        <label>Deploy.prototxt file:</label>
        <input class="form-control" type="text" placeholder
            ={{config['deploy']}} readonly><br>

        <label>Caffe model:</label>
        <input class="form-control" type="text" placeholder
            ={{config['weights']}} readonly><br>
    </div>
    <div class="col" style="margin-top: 25px">
        <form action="{{url_for('update_config')}}" method="
            post" enctype="multipart/form-data">
            <div class="form-group">
                <label for="deploy"><strong> Choose new
                    deploy prototxt:</strong></label><br>
                <input type="file" name="model" accept=".
                    prototxt" id="deploy" class="form-
                    control-file">
            </div>
            <div class="form-group">
                <label for="weights"><strong>Choose new
                    caffe model:</strong></label><br>
                <input type="file" name="model" accept=".
                    caffemodel" id="weights">
            </div>
            <input type="submit" value="Update" class="btn
                btn-outline-primary">
        </form>
    </div>
</div>
{% endblock %}
```

templates/upload.html

```
{% extends 'logged-layout.html' %}
{% block body %}
<style>
  .card-img-top {
    width: 100%;
    height: 15vw;
    object-fit: cover;
  }
</style>

{% if card %}
  <h2 class="display-4 text-primary">Result</h2>
  <hr>
  <div class="row">
    <div class="col d-flex justify-content-center">
        {% if '0' in session[' result '] %}
      <div class="card border-success" style="width: 18rem
          ;">
        <img class="card-img-top" src={{session['temp']}}
            alt="Card image cap">
        <div class="card-body text-success">
          <h5 class="card-title">Patient ID: {{pid}}</h5>
          <p class="card-text">
            <br>
            <strong> Normal Size </strong> : {{ '%0.2f' |
                format(prob[0] * 100 | float)}} %
            <br>
            <strong> Cardiomegaly </strong> : {{ '%0.2f' |
                format(prob[1] * 100 | float)}} %
          </p>
          {% else %}
          <div class="card border-danger" style="width: 18
              rem;">
            <img class="card-img-top" src={{session[ 'temp
                ']}} alt="Card image cap">
            <div class="card-body text-danger">
              <h5 class="card-title">Patient ID: {{pid}}</h5
                  >
              <p class="card-text">
                <br>
                <strong> Normal Size</strong> : {{ '%0.2f' |
                    format(prob[0] * 100 | float)}} %
                <br>
                <strong> Cardiomegaly </strong> : {{ '%0.2f'
                    | format(prob[1] * 100 | float)}} %
              </p>
              {% endif %}

              <div class="card-footer text-center">
                <a href="{{url_for('upload')}}" class="btn btn
                    -outline-success">Dismiss</a>
                <a href="{{url_for('consult')}}" class="btn
                    btn-outline-danger">Consult</a>
```

```
            </div>
          </div>
        </div>
    </div>
    <div class="col" style="padding-top: 10%">
        {% if '0' in session['result'] %}
        <h4 class="lead text-success">
          The patient's x-ray shows <strong>little to no sign
              of cardiomegaly</strong>. <br>
          The patient may opt to subject for further
              examinations which might include:
          <br>
        </h4>
        <p class="lead text-secondary">
          <span data-feather="check-square"></span> A
              full physical exam<br>
          <span data-feather="check-square"></span> A
              second opinion<br>
          <span data-feather="check-square"></span>
              Blood tests
        </p>

        {% else %}
        <h4 class="lead text-danger">
          The patient's x-ray shows <strong>signs of
              cardiomegaly</strong>. <br>
          The patient may opt to subject for further
              examinations which might include: <br>
        </h4>
        <p class="lead text-secondary">
          <span data-feather="check-square"></span> A
              full physical exam <br>
          <span data-feather="check-square"></span>
              Blood tests <br>
          <span data-feather="check-square"></span>
              Stress Test
        </p>

        {% endif %}
    </div>
  </div>
  <div class='col d-flex justify-content-center'>
  {% else %}
    <h2 class="display-4 text-primary">X-Ray Prediction</
        h2>
    <p class="lead text-secondary">
      Perform prediction in a single x-ray image. Upload a
      <i>.png</i> image file as an input to the system.
      <br> The system will output the percentage of patient's
          risk for cardiomegaly.
      <br> If you have a question about the result, you can hit
      <i>Consult</i> and compose your message to the doctor.
    </p>
    <hr>
  <div class='col d-flex justify-content-center'>
      <form action="{{url_for('upload')}}" method="post"
          enctype="multipart/form-data">
        <span class="btn btn-default">
          <input type="file" name="photo" accept=".png">
        </span>
        <input type="submit" value="Upload" class="btn btn
            -outline-primary">
      </form>
{% endif %}

  </div>

  {% endblock %}
```

templates/includes/_formhelpers.html

```
{% macro render_field(field) %}
  {{ field(**kwargs)|safe }}
  {% if field.errors %}
    {% for error in field.errors %}
    <div class="alert alert-danger" role="alert">
      <small id="passwordHelp" class="text-danger">
        <span class="help-inline">{{ error }}</span>
      </small>
    </div>
    {% endfor %}
  {% endif %}
{% endmacro %}
```

templates/includes/_messages.html

```
{% with messages = get_flashed_messages(with_categories=true)
    %}
  {% if messages %}
    {% for category, message in messages %}

      <div class="alert alert-{{category}} alert-dismissible
          fade show">
        {{ message }}
        <button type="button" class="close" data-dismiss="
            alert" aria-label="Close">
          <span aria-hidden="true">&times;</span>
        </button>
      </div>
    {% endfor %}
  {% endif %}
{% endwith %}
```

templates/includes/_navbar.html

```
<div>
  <nav class="navbar navbar-expand-lg navbar-light rounded
      fixed-top">
    <a class="navbar-brand" href="/">
      <span data-feather="activity"></span>HeartSmart
    </a>

    <div class="collapse navbar-collapse" id="
        navbarsExampleDefault">
      <ul class="navbar-nav mr-auto">
        <li class="nav-item">
          <a class="nav-link" href="/">Home
            <span class="sr-only">(current)</span>
          </a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="/about">About</a>
        </li>
      </ul>
      <ul class="navbar-nav navbar-right">
        {% if session.logged_in %}
        <li class="nav-item">
          <a class="nav-link" href='/dashboard'>{{session['
              name']}}
            <span class="sr-only">(current)</span>
          </a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="/logout">Logout
            <span class="sr-only">(current)</span>
          </a>
        </li>
        {% else %}
        <li class="nav-item">
          <a class="nav-link" href='/register'>Register
            <span class="sr-only">(current)</span>
          </a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="/login">Login
            <span class="sr-only">(current)</span>
          </a>
        </li>
        {% endif %}

      </ul>
    </div>
  </nav>
</div>
```

XI. Acknowledgement

I would like to take this opportunity to thank everyone who helped me make this happen, in making my 5-year UP experience be awesome.

First and foremost, I would like to thank Mama and Papa for always being there for me. Thank you for waking me up even if most of the times I'll just fall back asleep and ending up waking up late. Thank you for supporting my decisions and guiding me when sometimes it's a wrong one. Thank you for waking up and checking up on me during my sleepless nights. Thank you for everything. I'll forever be indebted to both of you.

To my Ate, thank you for buying me laptop when I needed one, for giving me extra money for my gala, and just being there when I need someone to talk to.

To my adviser Sir Marvin, thank you for being patient enough to explain the algorithm for my SP over and over even if I've asked for n-times. Thanks for helping me complete this SP.

To Jhoanna, my dorm buddy, food buddy, body buddy, CPP buddy. Thank you for supporting me and being my closest friend. I'll always remember our 2 am Mcdo BFF fries sessions, art making contests (lol), food critic era, random gala, pretentious jogging sessions. Papayat na tayo soon?

To Rhea, my SYDP buddy & Ferdie babies. Thank you for all the laughs and chismisan, to heart-to-heart talk on Sunken Garden and Intra, to random karaoke. Thank you for being an awesome friend and making my college experience less boring. Hoping to see your Sioselig Caffe soon in Australia.

To MASTERS 2018, thank you Abe, Faye, and Naiza. Thank you for making this SP experience less stressful. Thanks for always being there to support each other when we needed one. I'll never forget our sleepless nights together. SB, CBTL, Coreon, Boulangerie, Mr. Park's, Tim Hortons, Chocolat, Tambayan, Cafe Esso, etc.. Napuntahan na ata natin lahat ng coffee shops sa Manila at QC. Where should we

go next?

To Block 12, thank you for accepting me and treating me as a part of the block.

To DPSM, especially to Tita A, Ate She, and Ate Eden. I'll never forget my SA days. Thank you for helping me throughout my stay here in UP, for helping me process my papers, for helping me solve my countless SAIS problems, for your advice just to help me graduate on time (relative to my stay in UPM). I'll forever be thankful.

To my professors, thank you for imparting your knowledge and wisdom. This helped me become a better version of myself.

Lastly, I would like to thank you for not giving up, congratulate you for making it this far, and welcome you to the real world. Congrats, Edward.