**Pnemonia Detection using Convolutional Neural Network**

university of north texas

Department of computer science

software devepment for artificial intelligence

Project Proposal

**Pnemonia Detection using Convolutional Neural Network**

**by**

**Gopireddy Lakshmi Keerthi (Student ID: 11653876)**

*LakshmiKeerthiGopireddy@my.unt.edu*

**Major:** Artificial Intelligence

**Indhu Sri Krishnaraj (Student ID: 11661746)**

*IndhuSriKrishnaraj@my.unt.edu*

**Major:** Artificial Intelligence

**Vishal Rachuri (Student ID:** **11658243)**

*vishalrachuri@my.unt.edu*

**Major:** Artificial Intelligence

**Jyothika Vollireddy (Student ID: 11652684)**

*jyothikavollireddy@my.unt.edu*

**Major:** Cyber Security

**Pavani Jaya Keerthana Chittedi (Student ID: 11672192)**

*pavanijayakeerthanchittedi@my.unt.edu*

**Major:** Cyber Security

**Meeting Schedule and Collaboration**

We are a group of five students from various disciplines with varying academic schedules. Given the necessity for team cooperation, we have opted to meet virtually through Microsoft Teams/Google Meet or in person at Willis Library, depending on availability at the times listed below:

Monday: 6:00 PM to 7:00 PM

Thursday: 10:00 AM to 12:00 PM

Friday: 11:00 AM to 1:00 PM

We're utilizing these times to talk about how specific activities are progressing and to understand how one task is dependent on another. We made certain that the tasks and duties were distributed in such a way that each task was handled by two or more people. This allows us to brainstorm different ideas and strategies, resulting in a better learning curve and exposure.

We are using the below-mentioned platforms for smooth communication and secure information storage:

*SharePoint:* <https://myunt.sharepoint.com/:w:/r/sites/SDforAI-Project1>

Please note that the links have limited access. Drop an email to any team member for access.

**Individual Roles**

We divided the project into numerous tasks, and the work distribution per owner is summarized below:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tasks** | **Keerthi** | **Indhu** | **Vishal** | **Jyothika** | **Keerthana** |
| *Understanding the problem statement* | Yes | Yes | Yes | Yes | Yes |
| *Understanding Convolutional Neural Networks* | Yes | Yes | Yes | Yes | Yes |
| *Dataset Selection* | - | Yes | Yes | - | - |
| *Project Proposal* | Yes | Yes | Yes | Yes | Yes |
| *Domain Understanding* | Yes | Yes | Yes | Yes | Yes |
| *Loading the Dataset* | - | - | - | Yes | Yes |
| *Exploratory Data Analysis* | Yes | - | - | Yes | - |
| *Data Pre-processing* | - | - | Yes | - | Yes |
| *Data Augumentation* | Yes | Yes | - | - | - |
| *Model Training using CNN* | - | - | Yes | - | Yes |
| *Model Prediction* | - | Yes | - | Yes | - |
| *Model Evaluation* | Yes | - | - | - | Yes |
| *Model Re-training/ Parameter tuning* | Yes | - | Yes | - | - |
| *Model Optimization* | Yes | - | Yes | - | - |
| *Model Evaluation and Final Model Saving* | - | Yes | - | Yes | - |
| *Designing an User Interface* | - | Yes | - | - | Yes |
| *Integration of Saved Model and UI* | - | Yes | Yes | - | - |
| *User Interface Performance Check* | Yes | - | - | Yes | - |
| *Project Documentation* | Yes | Yes | Yes | Yes | Yes |
| *Project Presentation* | Yes | Yes | Yes | Yes | Yes |

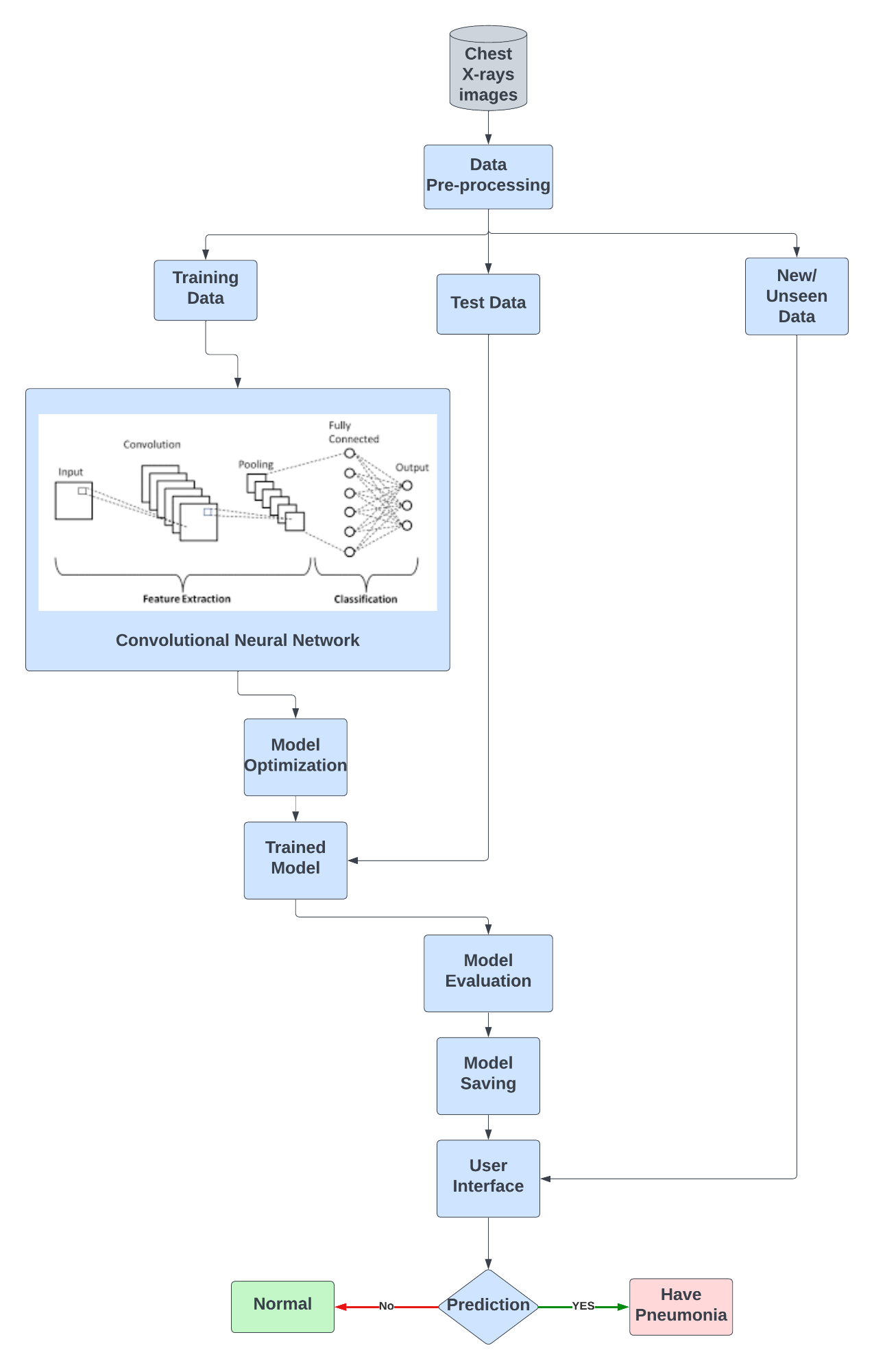
**Project Abstract**

Medical image classification is a critical task in healthcare that involves analyzing images from medical devices to help doctors make diagnoses and treatment decisions. A large dataset of medical images will be used to train and validate the model, and its performance will be tested on a separate set of images. In this project, we will be predicting whether a person has pneumonia or not given an X-Ray image using Convulutional Neural Networks. The project will entail gathering and preparing the dataset, designing and constructing the CNN architecture, training and fine-tuning the network, and assessing its performance. The project's main goal is to create a dependable and accurate tool that can be used in a clinical setting to improve patient outcomes.

To achieve the goals of this project, several steps will be followed. Firstly, a large dataset of medical images will be collected and labeled for training and validation purposes. The dataset will be carefully curated to ensure that it is diverse and representative of the range of medical conditions that will be classified by the CNN. To improve the CNN's ability to generalize to new, unseen images, preprocessing steps such as resizing, normalization, and data augmentation will be applied to the dataset. To balance performance and computational efficiency, the number and size of convolutional layers, pooling layers, and fully connected layers will be optimized. Standard optimization techniques such as backpropagation and stochastic gradient descent will be used to train the network, and hyperparameters such as learning rate, optimizer, and number of layers will be fine-tuned to improve the CNN's performance.

To assess the CNN's ability to generalize to new, unseen images, its performance will be evaluated on a separate set of images that were not used during training. To provide a comprehensive assessment of the CNN's performance, the evaluation will consider metrics such as accuracy, precision, recall, and F1 score. Based on the evaluation results, the network will be fine-tuned further if necessary. The stored model is then utilized to predict the disease using real-time X-rays on a Graphical User Interface, or Website.

**Visual Demonstration – Pnemonia Detection using CNN**



**Project Design and Technology Stack**

The technology stack for implementing a pneumonia prediction system can include the following:

1. ***Programming language:*** The CNN model may be built using Python as the programming language. TensorFlow, Keras, and PyTorch are three Python machine learning and deep learning libraries that may be used to develop CNN models. For this project, we will be using Keras considering its easy-to-use interface for building and training deep learning models.
2. ***Server-side configurations:*** We will need a powerful server or a cloud-based platform with GPUs to train the CNN model efficiently. We can use cloud-based platforms such as Google Colab.
3. ***Client-side hardware/software:*** To train the CNN model, we'll need a machine with a powerful CPU and GPU. We will also need to install the relevant libraries and APIs on the Computer in order to build and train the CNN model.
4. ***Dataset Requirements****:* We'll require a set of chest X-ray photos with pneumonia and non-pneumonia instances labeled. We are utilizing the publicly available Kaggle's Pneumonia dataset.
5. ***CNN Architectural Requirements:*** We will need to design the CNN architecture for the pneumonia prediction task. We can use pre-trained CNN models such as VGG, ResNet, or Inception as a starting point and fine-tune the model for our specific task.
6. ***Model Deployment:*** Once the CNN model is trained and evaluated, we can deploy it in a web application or a mobile app. We can use web frameworks such as Flask or Django for building the web application or mobile app development frameworks such as React Native or Flutter for building the mobile app.

Additionally, medical image classification using CNNs requires expertise in both computer vision and medical imaging. It is important to ensure that the model is properly validated and clinically relevant before deploying it in a real-world setting.

**Workflow**

Considering multiple levels of developing a machine learning model, we have come up with **10** milestones that will help us to evaluate the progress of the project:

1. Understanding the problem statement
2. Understanding Convolutional Neural Networks
3. Dataset Selection
4. Project Proposal
5. Domain Understanding
6. Data Preparation:
   1. Loading the Dataset
   2. Exploratory Data Analysis
   3. Data Pre-processing
   4. Data Augumentation
7. Model Training:
   1. Model Training using CNN
   2. Model Prediction
   3. Model Evaluation
   4. Model Re-training/ Parameter tuning
   5. Model Optimization
   6. Model Evaluation and Final Model Saving
8. User Interface Creation:
   1. Design an User Interface
   2. Integration of Saved Model and User Interface
   3. User Interface Performance Check
9. Project Documentation
10. Project Presentation

**References**

**Datasets:**

<https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia>

<https://www.kaggle.com/datasets/tolgadincer/labeled-chest-xray-images>

<https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database>

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8423280/>

<https://ieee-dataport.org/documents/chest-xray-pneumonia>

**Research Papers:**

<https://github.com/topics/pneumonia-detection>

<http://www.ijpe-online.com/EN/10.23940/ijpe.22.05.p8.380386>

<https://www.researchgate.net/publication/352692651_Pneumonia_Detection_from_Chest_X-ray_Images_Based_on_Convolutional_Neural_Network>

<https://ieeexplore.ieee.org/document/8741582>

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9759647/>

**Algorithm Papers:**

<https://cs.nju.edu.cn/wujx/paper/CNN.pdf>

<https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3833453>

<https://medium.com/@monocosmo77/research-papers-based-on-r-cnn-a-type-of-convolutional-neural-network-in-deep-learning-13538ca15b0d>

<https://www.researchgate.net/publication/333242381_Convolutional_Neural_Network_Layers_and_Architectures>