Project Proposal

Group: BDA-2102
Students: Zhauymbayeva Madina, Serikkazy Ulpan
Project name: Pneumonia detection using X-ray images

- Problem or idea description

Pneumonia is one of the very contagious diseases, the so-called "Lung Infection" that has affected the lives of millions of people around the world. You're more likely to get pneumonia if you have asthma, chronic obstructive pulmonary disease (COPD) or heart disease. Most often, this disease causes respiratory symptoms that can be very similar to shortness of breath, acute respiratory viral infections or tachycardia. Pneumonia can range in seriousness from mild to life-threatening. Pneumonia can be quickly and accurately diagnosed using computed tomography (CT) and chest X-ray (CXR). However, since it takes a long time and is very prone to human error, identifying an infection manually using a radio image is quite difficult. Our goal is to create an image recognition model that will allow us to determine the presence of Pneumonia from an X-ray image of the patient's lungs.

- Background information on the problem or idea

Due to the high contagiousness of Pneumonia, it is important to screen, diagnose and isolate patients as soon as possible in order to stop the transmission of the disease and accelerate their proper treatment. Medical imaging techniques such as computed tomography and CXR are often used in the diagnosis of Pneumonia infection and give accurate results. However, given the recent onset of the disease and parallels with other lung diseases such as Bronchitis, accurate interpretation of the results based on images is difficult. Convolutional Neural Network (CNN) serves as a key building component in most DL-based models, and each layer extracts Pneumonia-related characteristics that can be used to distinguish Pneumonia images from non-Pneumonia images. Since automatic analysis of CNN characteristics is possible, a classification of Pneumonia based on deep neural networks is usually used.
Symptoms of Pneumonia:

1. Chest pain
2. Cough
3. Fever
4. Fatigue

- Available solutions with links are provided below

Author of this model used a custom deep convolutional neural network and retraining a pre-trained “InceptionV3" model to identify pneumonia from x-ray images. For retraining, he removed the output layers, freezed the first few layers, and fine-tuned the model for two Pneumonia and Normal classes.
https://github.com/anjanatiha/Pneumonia-Detection-from-Chest-X-Ray-Images-with-Deep-Learning

Author of this project divided the train set into training and validation data. He solved the problem of data imbalance using stratified class. Training and validation set split changed from 99:1 ratio to 90:10 ratio.
https://www.kaggle.com/code/wirachleelakiatiwong/pneumania-classification-transfered-cnn/notebook

In this project, the author compared pre-trained models such as VGG16, CNN_2, ResNet, InceptionNet, DenseNet with their mean absolute error. Also, the author built his own CNN model that showed 91% of accuracy.
https://github.com/0xpranjal/Pneumonia-Detection-using-Deep-Learning

- How to get the data?

The model will use a dataset from Kaggle and this dataset is divided into train, test and validation data. Overall, there are 5863 X ray images (JPEG) and each of the 3 folders are breached into 2 subfolders (Normal / Pneumonia), containing first chest X ray images of a healthy person and second is for the X-rays of a person with pneumonia.
https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia

- Brief description of your solution

We will build our model in Python using libraries such as keras and tensorflow for building and testing models, and matplotlib for graphical representation. First of all, we will prepare our dataset, by defining directories and files (train, test and validation). The model will have a training set "train_images", "train_labels" to learn and a validation set to determine the parameters of the model. We will visualize the dataset to see what a normal person's X-ray and X-ray of a person with pneumonia looks like with matplotlib.pyplot. Then, we will build a neural network by configuring layers, compiling the model, and training hyperparameters. After which evaluation of accuracy, making and verifying of predictions will take place.

- Tech stack that will be used

Programming language: Python
Tools / IDE: Google Colab
Libraries: Tensorflow, Keras, Numpy, Matplotlib, Seaborn, Sklearn, Gradio

At first, the IDE was Jupyter Notebook, but we changed it to Google Colab because it provides free computational power, built-in libraries and we can work on the same notebook in real-time.