

# Detection of Lung Infection.

Project 1

## DESCRIPTION

Artificial Intelligence has evolved a lot and is currently able to solve problems that are very complex and require human specialization. One such area is healthcare.

A lot of research happens every day to use deep learning for the betterment of humanity, and one such is healthcare.

### Objective:

To build a model using a convolutional neural network that can classify lung infection in a person using medical imagery

### Dataset Description:

The dataset contains three different classes, including healthy, type 1 disease, and type 2 disease.

- **Train folder:** This folder has images for training the model, which is divided into subfolders having the same name as the class.
- **Test folder:** This folder has images for testing the model, which is divided into subfolders having the same name as the class.

### Following operations should be performed using Keras or PyTorch or Torch vision-

1. Import the necessary libraries
2. Plot the sample images for all the classes
3. Plot the distribution of images across the classes
4. Build a data augmentation for train data to create new data with translation, rescale and flip, and rotation transformations. Rescale the image at 48x48
5. Build a data augmentation for test data to create new data and rescale the image at 48x48
6. Read images directly from the train folder and test folder using the appropriate function

### Build 3 CNN model with:

1. **CNN Architecture:**
  1. Add convolutional layers with different filters, max pool layers, dropout layers, and batch normalization layers
  2. Use Relu as an activation function

3. Take the loss function as categorical cross-entropy
4. Take rmsprop as an optimizer
5. Use early stopping with the patience of two epochs and monitor the validation loss or accuracy
6. Try with ten numbers epoch
7. Train the model using a generator and test the accuracy of the test data at every epoch
8. Plot the training and validation accuracy, and the loss
9. Observe the precision, recall the F1-score for all classes for both grayscale and color models, and determine if the model's classes are good

## 2. **Transfer learning using mobile net:**

1. Prepare data for the pre-trained mobile net model, with color mode as RGB
2. Create an instance of a mobile net pre-trained model
3. Add dense layer, dropout layer, batch normalization layer on the pre-trained model
4. Create a final output layer with a SoftMax activation function
5. Change the batch size activation function and optimize as rmsprop and observe if the accuracy increases
6. Take the loss function as categorical cross-entropy
7. Use early stopping with the patience of two epoch and call back function for preventing overfitting
8. Try with ten numbers epoch
9. Train the model using a generator and test the accuracy of the test data at every epoch
10. Plot the training and validation accuracy, and the loss
11. Observe the precision, recall the F1-score for all classes for both grayscale
12. and color models, and determine if the model's classes are good

## 3. **Transfer Learning using Densenet121:**

1. Prepare the dataset for the transfer learning algorithm using Densenet121 with the image size as 224x224x3
2. Freeze the top layers of the pre-trained model
3. Add a dense layer at the end of the pre-trained model followed by a dropout layer and try various combinations to get an accuracy
4. Add the final output layer with a SoftMax activation function
5. Take loss function as categorical cross-entropy
6. Take Adam as an optimizer
7. Use early stopping to prevent overfitting

8. Try with 15 number of epoch and batch size with seven, also try various values to see the impact on results
9. Train the model using the generator and test the accuracy of the test data at every epoch
10. Plot the training and validation accuracy, and the loss
11. Observe the precision, recall the F1-score for all classes for both grayscale and color models, and determine if the model's classes are good

**Final step:**

1. Compare all the models on the basis of accuracy, precision, recall, and f1-score