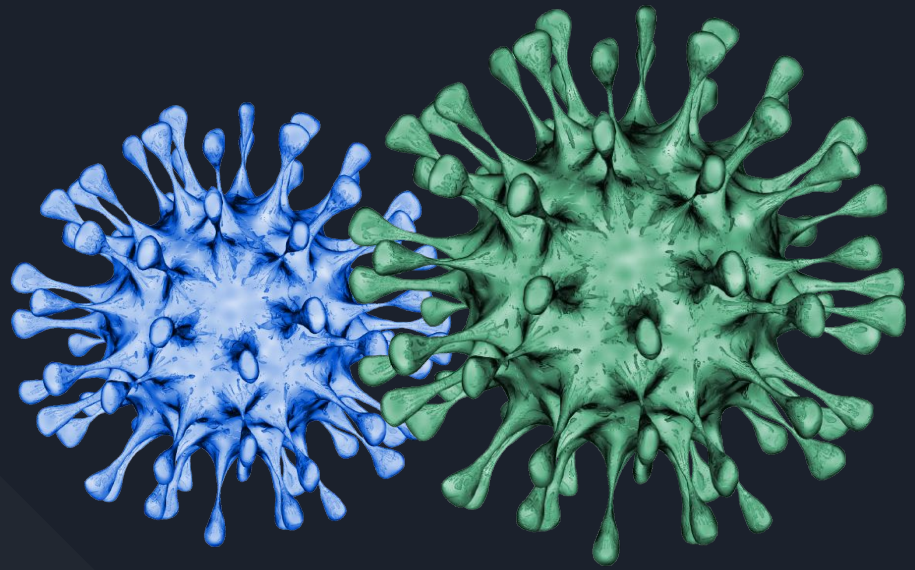
COVID-19 CLASSIFICATION USING CHEST X-RAY IMAGES



Presentation By-
Ankur Agrawal (20095010)
Vivek Agrawal (20095126)

CONTENTS

- PROJECT OBJECTIVE
- FEATURE OF PROJECT
- BLOCK DIAGRAM
- PROPOSED SYSTEM ARCHITECTURE
- CLASSIFICATION USING CNN AND FUZZY LOGIC
- SYSTEM IMPLEMENTATION
- RESULT
- CONCLUSION

PROJECT OBJECTIVE

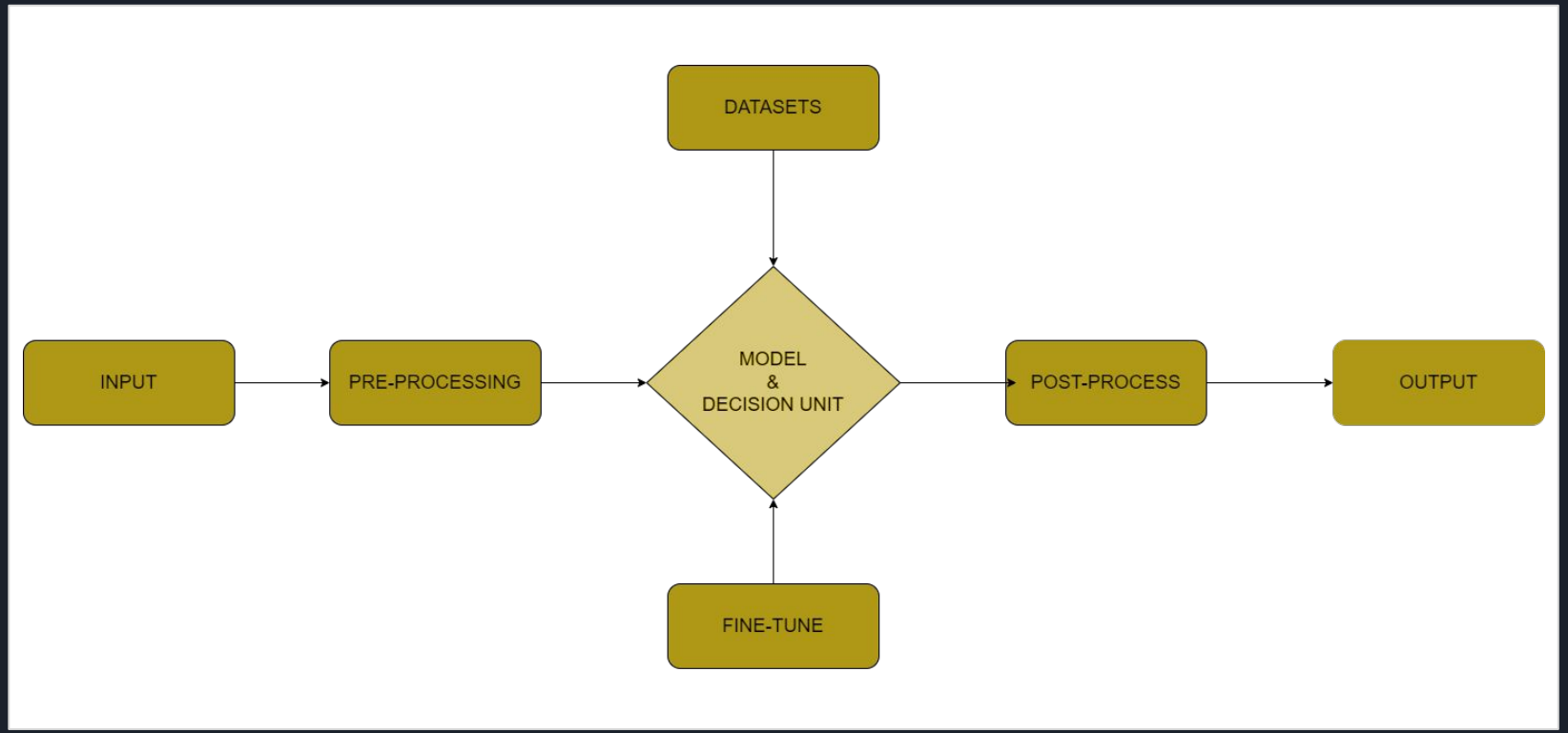
- The aim of the project is to develop a novel deep neural network based model for highly accurate detection of COVID-19 infection from the chest X- Ray images of the patients.

- The aim of our project is to make use of modern AI techniques to detect the COVID-19 patients using X-Ray images in an automated manner, particularly in settings where radiologists are not available, and help make the proposed testing technology scalable

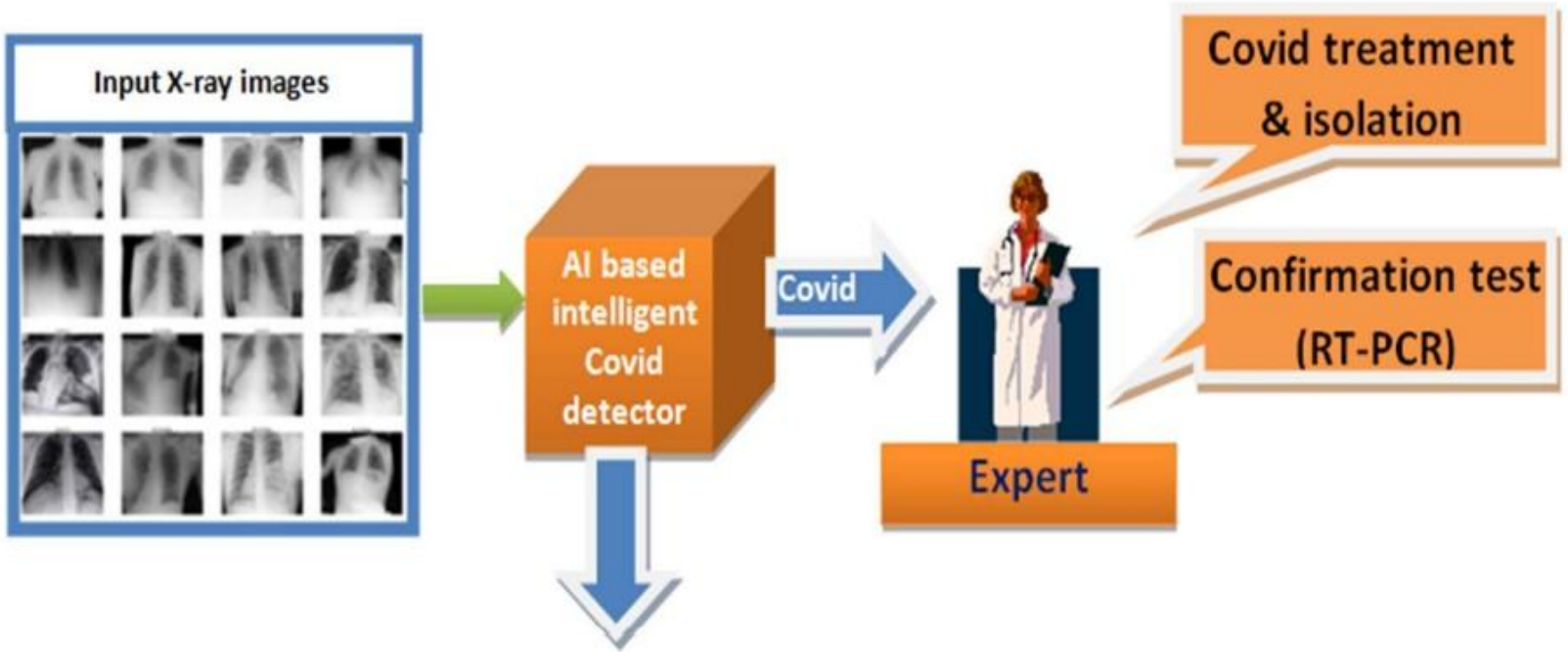
FEATURES OF PROJECT

- Early diagnosis is very essential for both early intervention to the patient and prevent the risk of transmission of the disease. In this Project we have presented a neural network architecture which will reduce overfitting.
- It can also be used in situation where the possibilities are insufficient whether in terms RT-PCR test or doctor. We have used a fuzzy neural network instead of taking a fully connected layer for the feature information.
- We all know that Convolutional neural network is an excellent feature extractor so integrating it with the fuzzy neural network will increase its accuracy in tests.

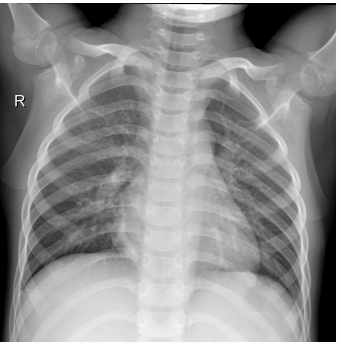
BLOCK DIAGRAM

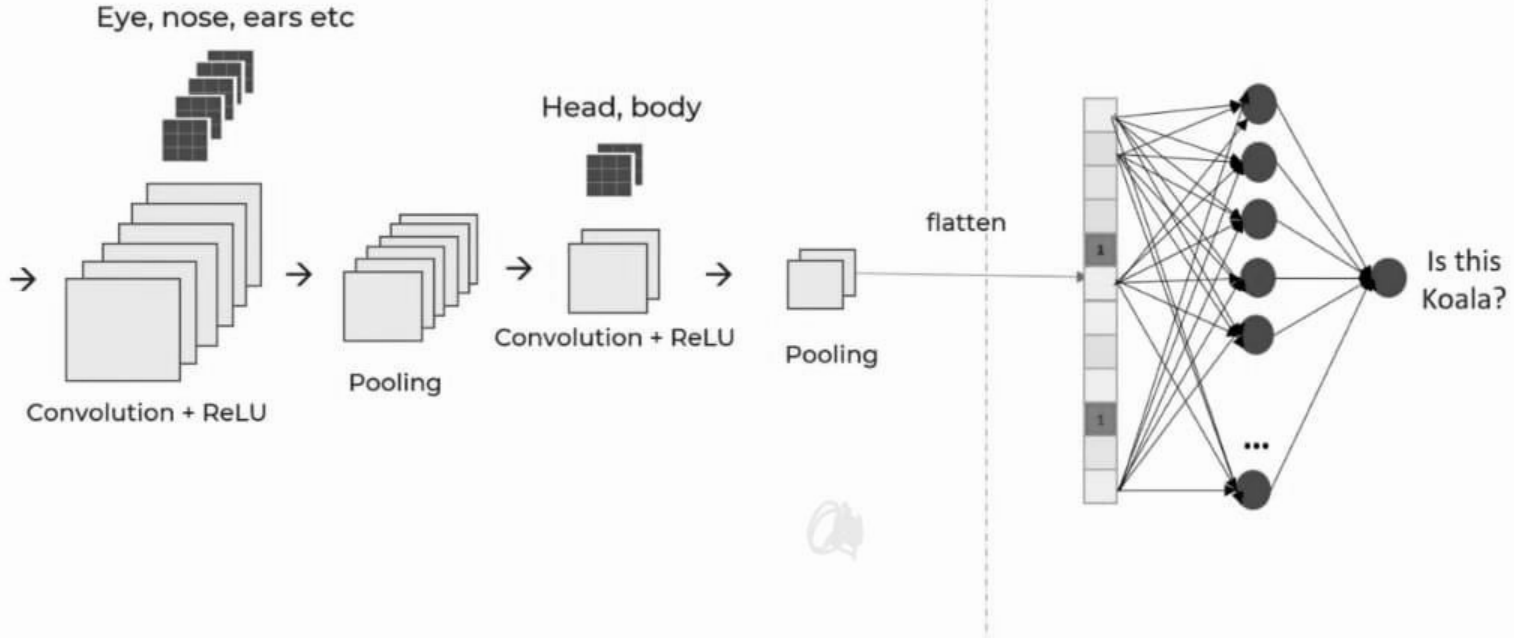


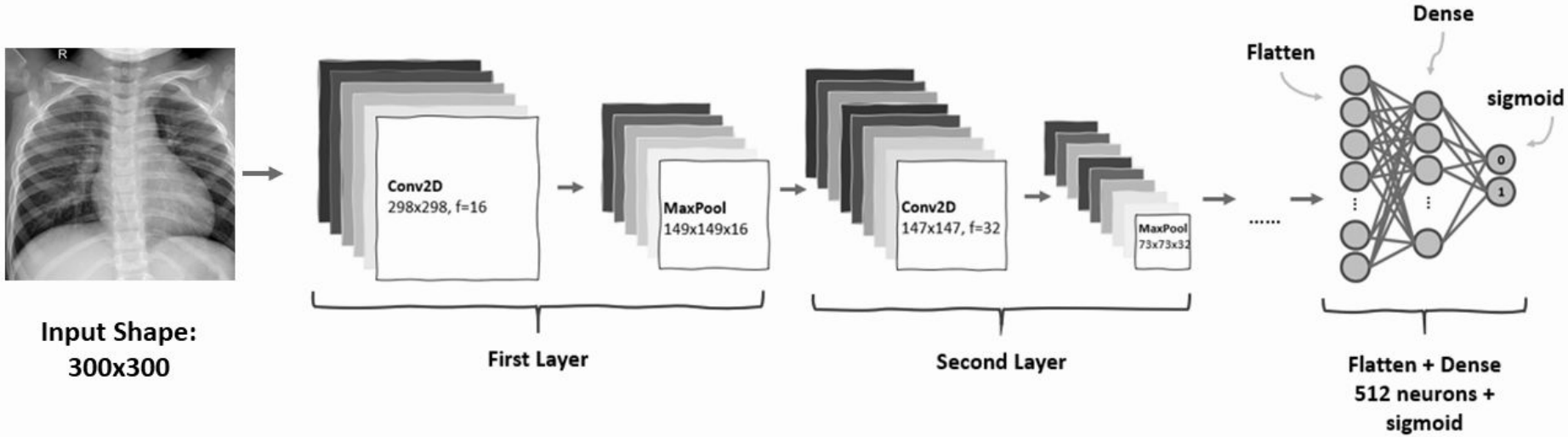
Proposed Systems Architecture



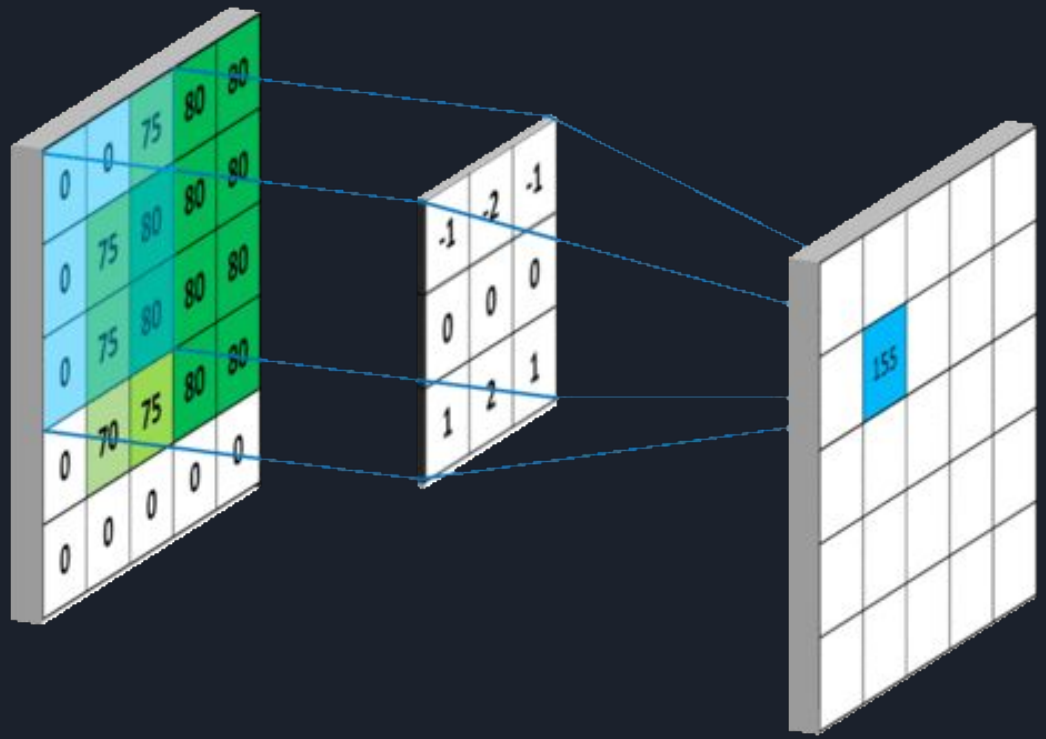
CLASSIFICATION USING CNN



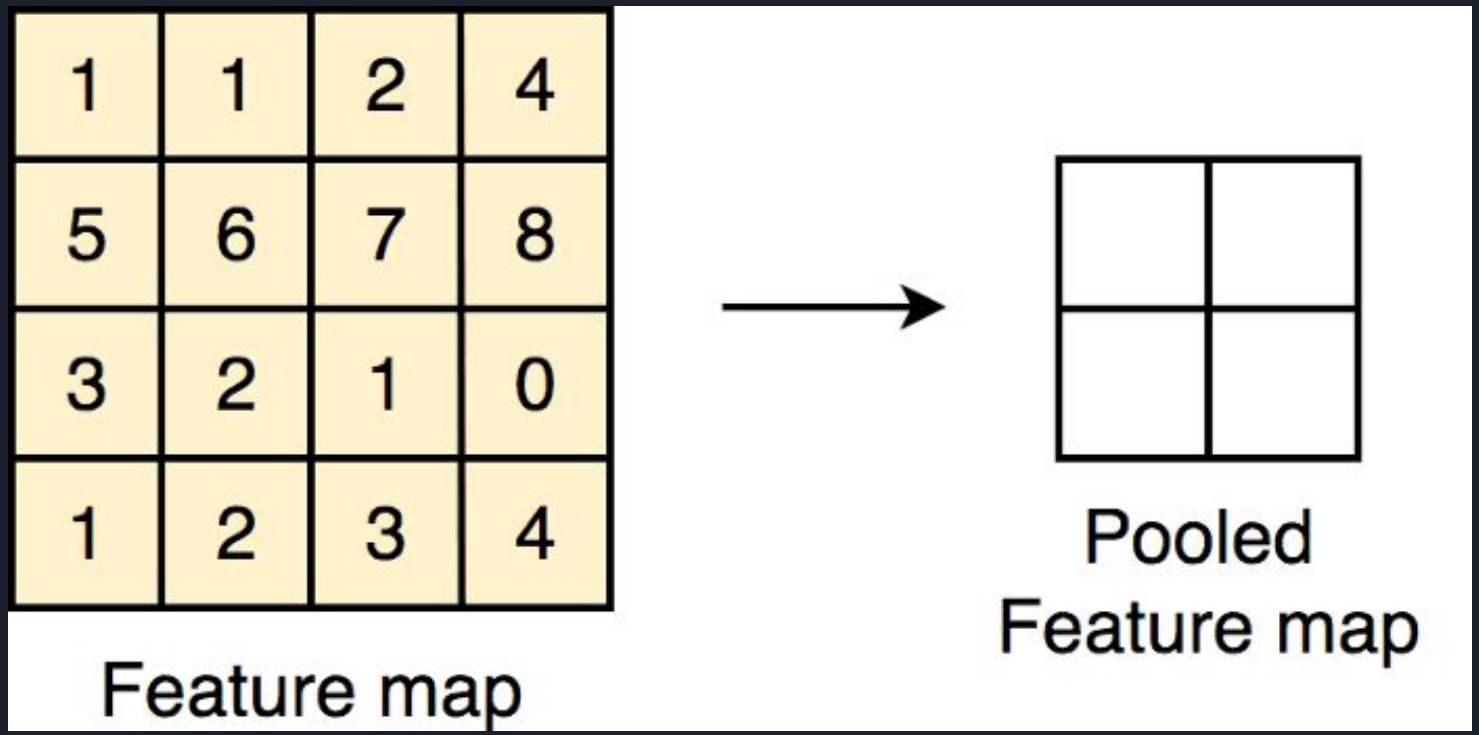



R
Input Shape:
300x300
Conv2D
298x298, f=16
MaxPool
149x149x16
First Layer
Conv2D
147x147, f=32
MaxPool
73x73x32
Second Layer
Flatten
Dense
sigmoid
0
1
.....
Flatten + Dense
512 neurons +
sigmoid

Convolutional Operation

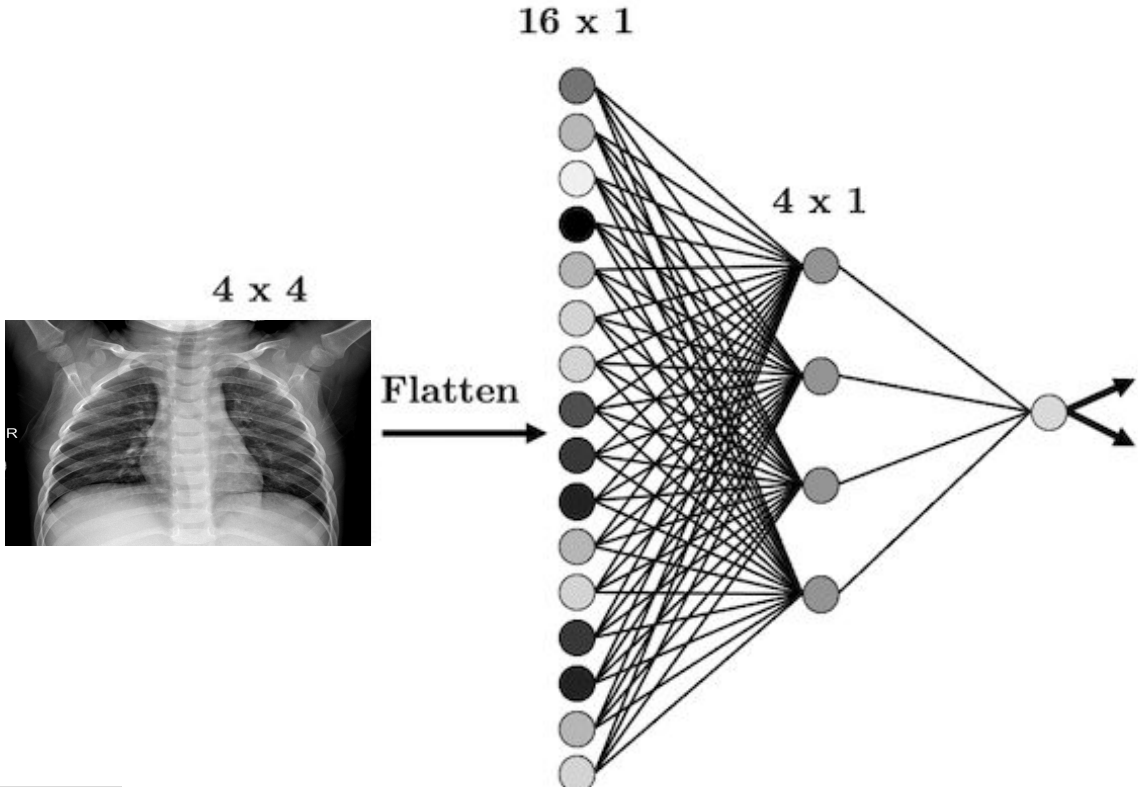


MAX POOLING



CLASSIFICATION

$$\text{Total Parameters} = (16 \times 4 + 4) + (4 \times 1 + 1) = 73$$



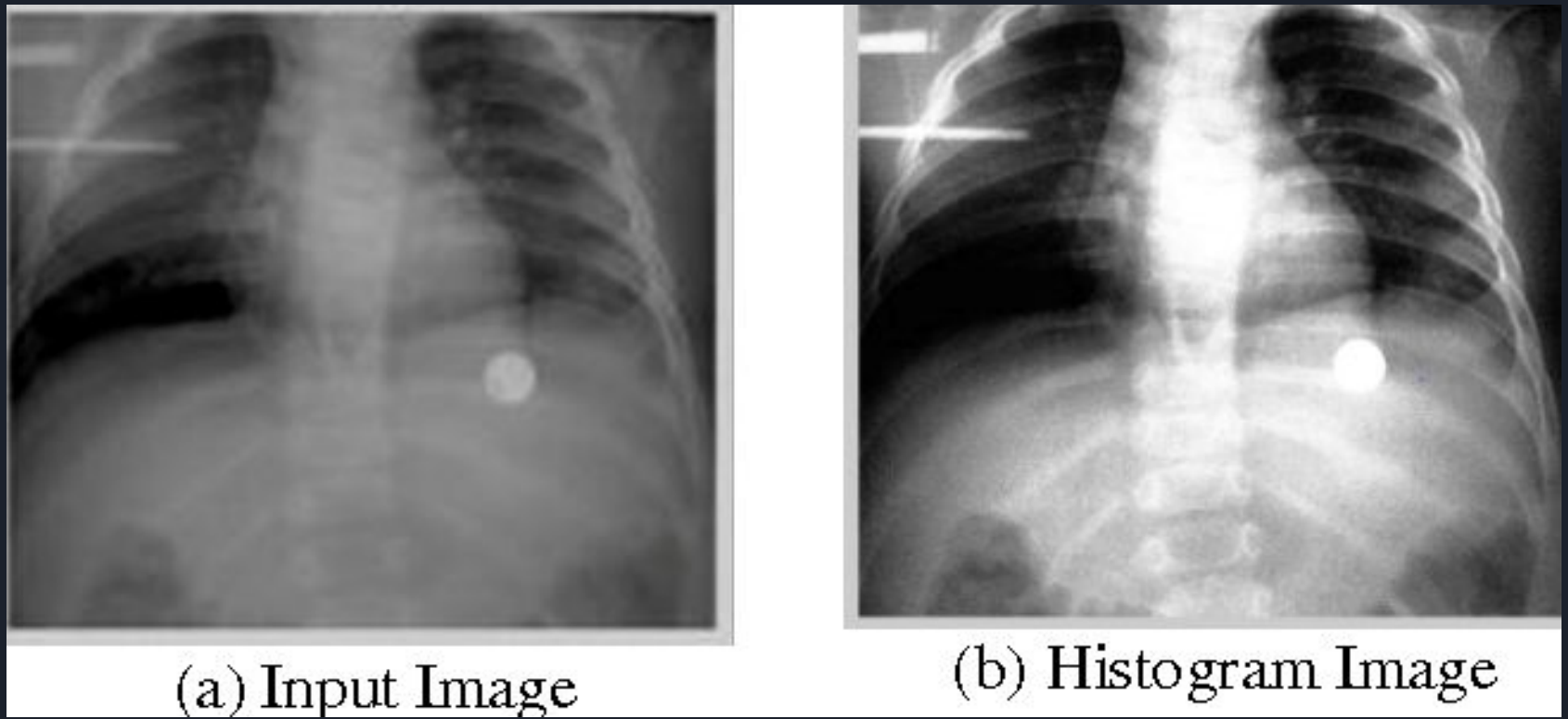
CLASSIFICATION USING FUZZY LOGIC

- Fuzzy logic is a method that was introduced to handle a range of values between TRUE and FALSE. This model of reasoning has helped us to reach closer to human reasoning. Most of the models which work on crisp data set only give output in one of the two forms: False (Value 0) or True (Value 1), Negative (Value 0) or Positive (Value 1) but what if there are other possibilities like partially True or partially False.
- A model which can produce the following possible decisions between FALSE and TRUE: Surely True, might be True, Not Sure about True or False, Might Be False, surely False. All these decisions can be achieved with the help of Fuzzy logic.
  - A. Fuzzy Based Image Processing
  - B. Fuzzy Based Classification on X-ray Images

Fuzzy Based Classification on X-ray Images

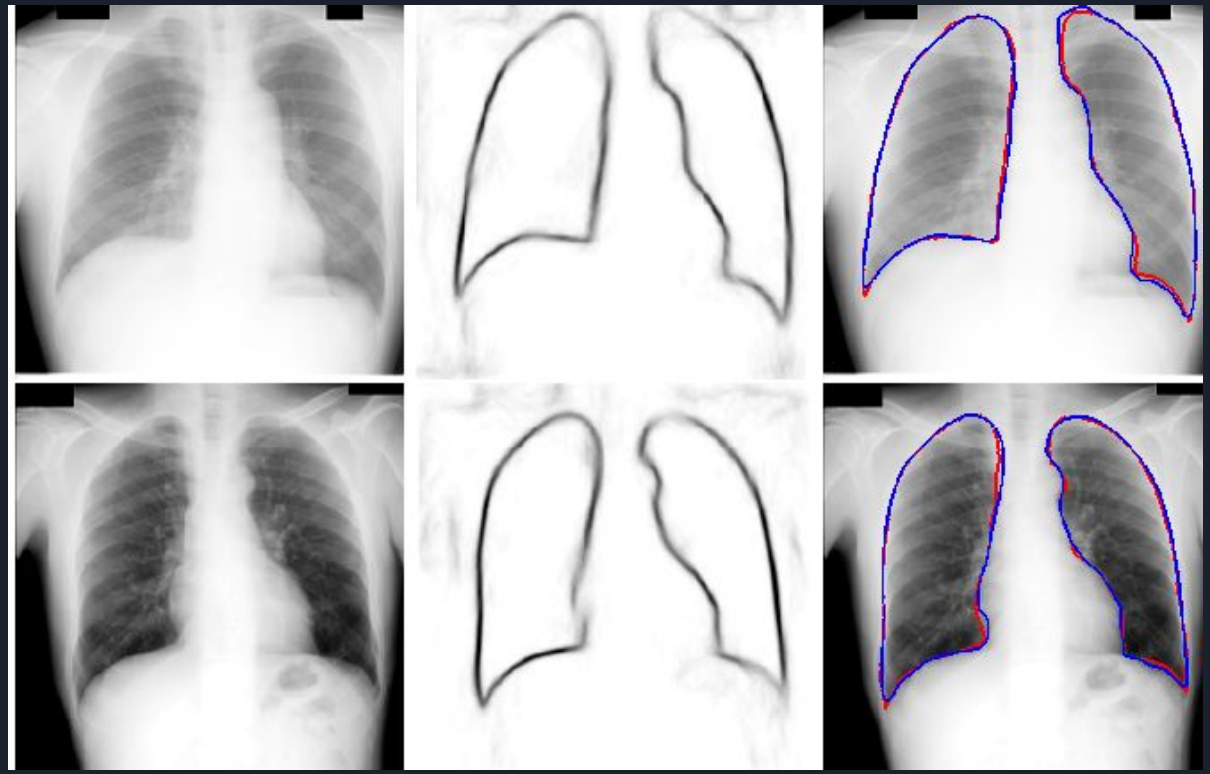
Feature Extraction:

- In pattern classification, we use features like color, shape, spatial location, and texture. It is not suitable to consider color and texture features for x-ray image classification. The reason is that x-ray images are gray-scale images, and their texture characteristics are remarkably similar. It is recommended to use combinations of texture and shape features together.
- Histogram Adjustment
- Noise Removal
- Edge Detection



(a) Input Image

(b) Histogram Image



Edge Detection

System Implementation and Algorithm

We have taken datasets that are publicly available. As we are doing binary classification, we have taken positive covid 19 X-ray images from the GitHub repository which is being updated by a group of doctors on a regular basis. As we wanted a balanced dataset therefore, we have taken the same number of normal X-ray images from Kaggle as we have taken from GitHub repository.

Link : GitHub - ieee8023/covid-chestxray-dataset: We are building an open database of COVID-19 cases with chest X-ray or CT images.

In [1]:
```
import pandas as pd
import os
from pathlib import Path
import shutil
import random
import math

import numpy as np
import matplotlib.pyplot as plt

import keras
from keras.layers import *
from keras.models import *
from keras.preprocessing import image
from tensorflow.keras.preprocessing import image as imager

import gradio as gr
from PIL import Image as im
```

In [4]:
```
FILE_PATH = Path('chestxray/metadata.csv').absolute()
IMAGE_PATH = Path('chestxray/images/').absolute()
```

In [ ]:
```
```

In [5]:
```
df = pd.read_csv(FILE_PATH)
```

In [6]:
```
df.head()
```

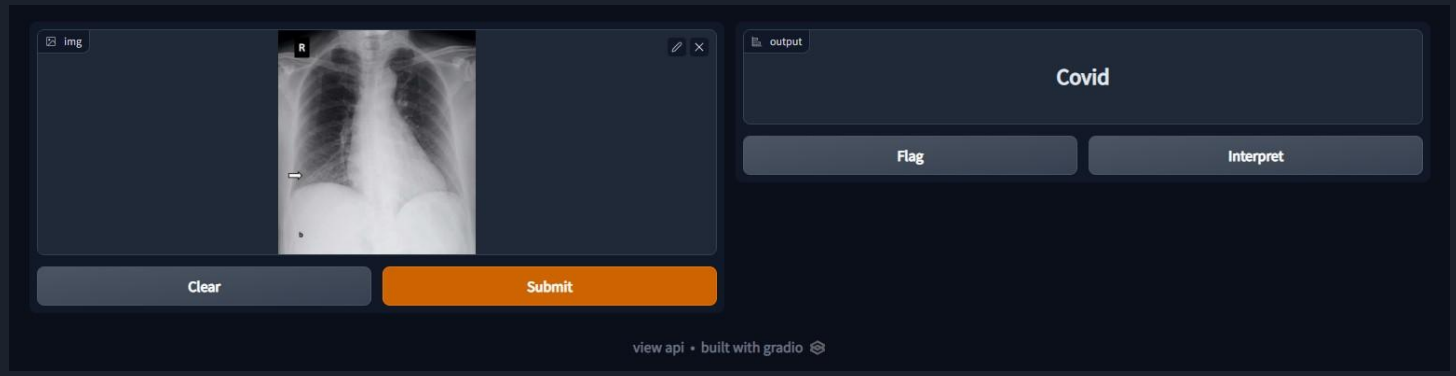
Out[6]:

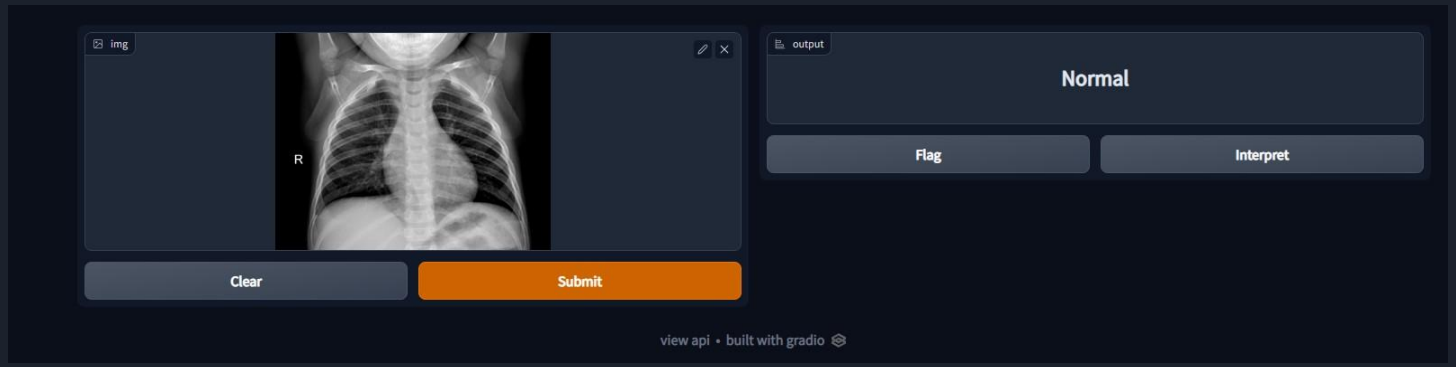
	patientid	offset	sex	age	finding	RT_PCR_positive	survival	intubated	intubation_present	went_icu	...	date	location	folder	
0	2	0.0	M	65.0	Pneumonia/Viral/COVID-19	Y	Y	N	N	N	...	January 22, 2020	Cho Ray Hospital, Ho Chi Minh City, Vietnam	images	20
1	2	3.0	M	65.0	Pneumonia/Viral/COVID-19	Y	Y	N	N	N	...	January 25, 2020	Cho Ray Hospital, Ho Chi Minh City, Vietnam	images	20
2	2	5.0	M	65.0	Pneumonia/Viral/COVID-19	Y	Y	N	N	N	...	January 27, 2020	Cho Ray Hospital, Ho Chi Minh City, Vietnam	images	20

In [3]:
```
model = Sequential()
model.add(Conv2D(32, kernel_size=(3, 3), activation='relu', input_shape=(224, 224, 3)))
model.add(Conv2D(64, kernel_size=(3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Conv2D(64, kernel_size=(3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Conv2D(128, kernel_size=(3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Flatten())
model.add(Dense(64, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(1, activation='sigmoid'))

model.compile(loss=keras.losses.binary_crossentropy, optimizer='adam', metrics=['accuracy'])
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 222, 222, 32)      896

 conv2d_1 (Conv2D)           (None, 220, 220, 64)      18496

 max_pooling2d (MaxPooling2D  (None, 110, 110, 64)     0
 )

 dropout (Dropout)           (None, 110, 110, 64)      0

 conv2d_2 (Conv2D)           (None, 108, 108, 64)      36928

 max_pooling2d_1 (MaxPooling  (None, 54, 54, 64)       0
 2D)

 dropout_1 (Dropout)         (None, 54, 54, 64)        0

 conv2d_3 (Conv2D)           (None, 52, 52, 128)       73856

 max_pooling2d_2 (MaxPooling  (None, 26, 26, 128)      0
 2D)
```

RESULTS


img
R
output
Covid
Flag
Interpret
Clear
Submit
view api • built with gradio


img
R
output
Normal
Flag
Interpret
Clear
Submit
view api • built with gradio

Conclusion

Early diagnosis is very essential for both early intervention to the patient and prevent the risk of transmission of the disease. In this Project we have presented a neural network architecture which will reduce overfitting. It can also be used in situation where the possibilities are insufficient whether in terms RT-PCR test or doctor.

Thank You !!!

ANY
QUESTIONS ?

References:

- https://arxiv.org/abs/2004.12823
- https://arxiv.org/abs/2003.11597
- https://ieeexplore.ieee.org/document/8482211
- https://www.sciencedirect.com/science/article/pii/S0957417422005073