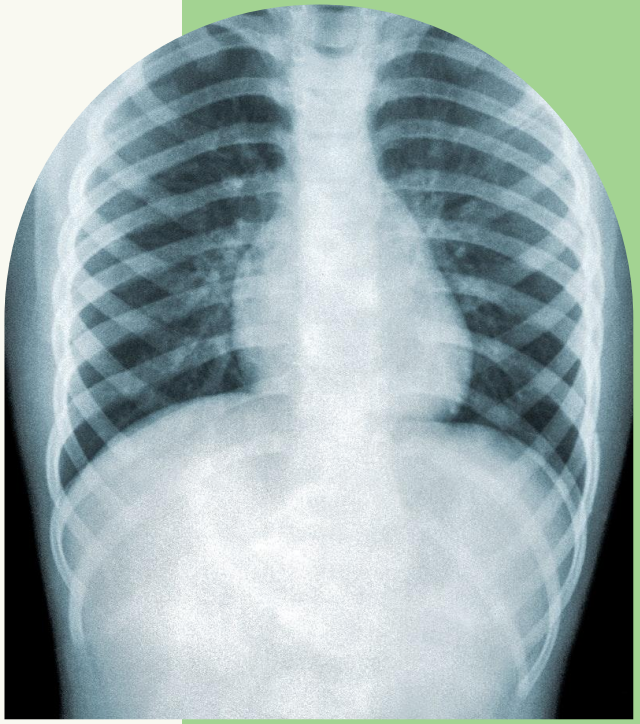
Pneumonia Detection Using Chest X-Ray Image

CSE - 444 (Group - 09)



Group Members

Tasneem Mubashshira	201814054
Ifath Ara	201814060
Fariha Fardina Amin	201814061

Presentation Outline

01
Problem Definition and Proposed Solution

02
Dataset

03
Model

04
Training

Presentation Outline

05

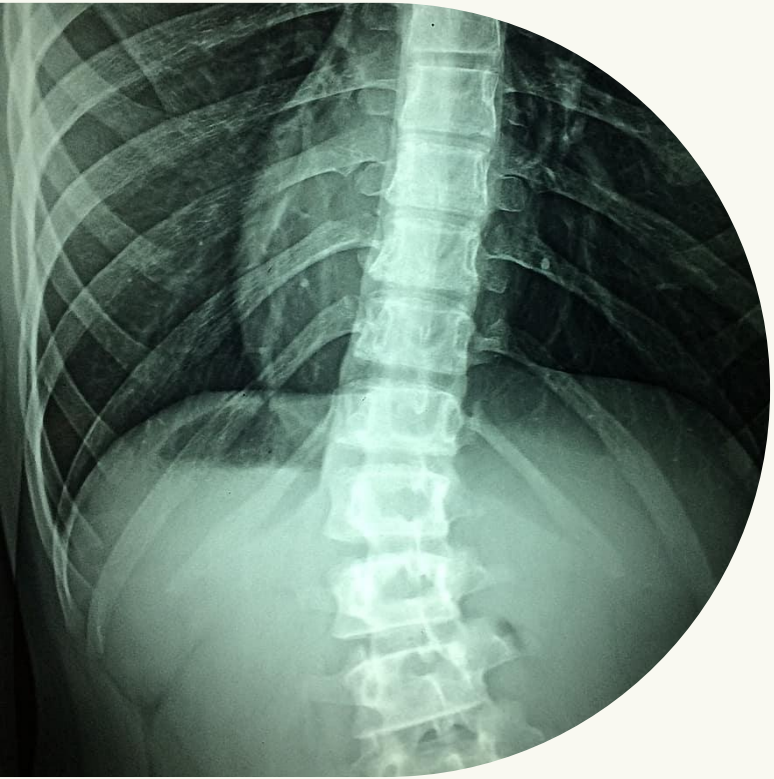
Evaluation

06

Limitations

07

Running the model on a data point



01

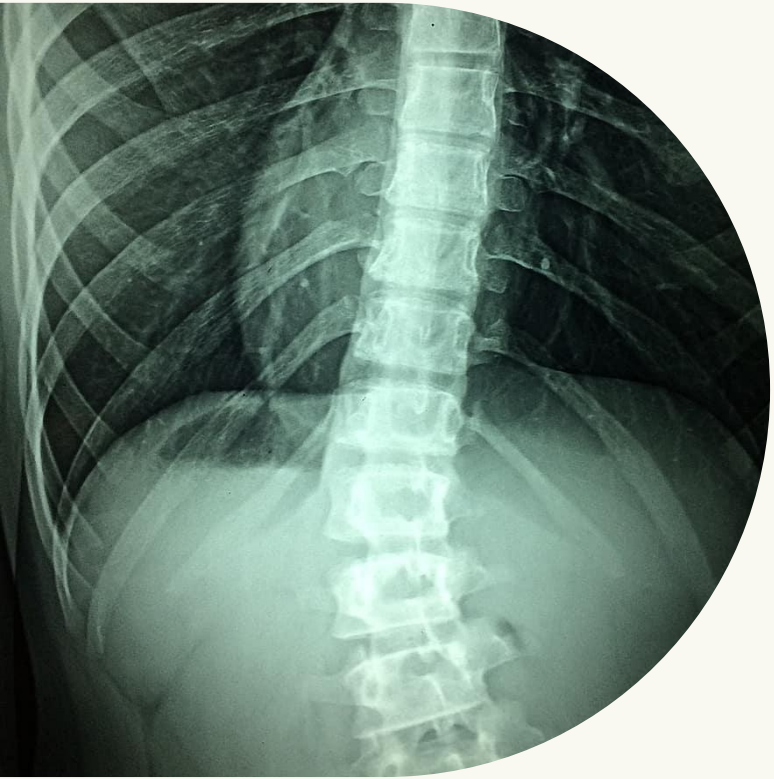
Problem Definition and Proposed Solution

Problem Definition and Proposed Solution

- Pneumonia is one of the major illnesses in children and aged humans due to the Infection in the lungs.
- Analysis of pneumonia is necessary to prepare for a possible treatment procedure to regulate and cure the disease.
- Chest X-ray imaging is the most frequently used method for diagnosing pneumonia.
- However, the examination of chest X-rays is a challenging task and is prone to subjective variability.
- In this study, we developed a deep learning based diagnosis system for automatic pneumonia detection using chest X-ray images.

Objectives

- Using manual preprocessing techniques and monitoring the model performance.
- Creating customized layers of CNN to compare the performance between them.
- Using famous CNN architecture's pretrained models and compare results between them.
- Doing a comparative analysis between all the techniques used.



02

DataSet

DataSet Description

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

Kaggle Dataset

(organized in 3 folders
- Train, Test,Validation)

5836 X-RAY Images (jpeg)

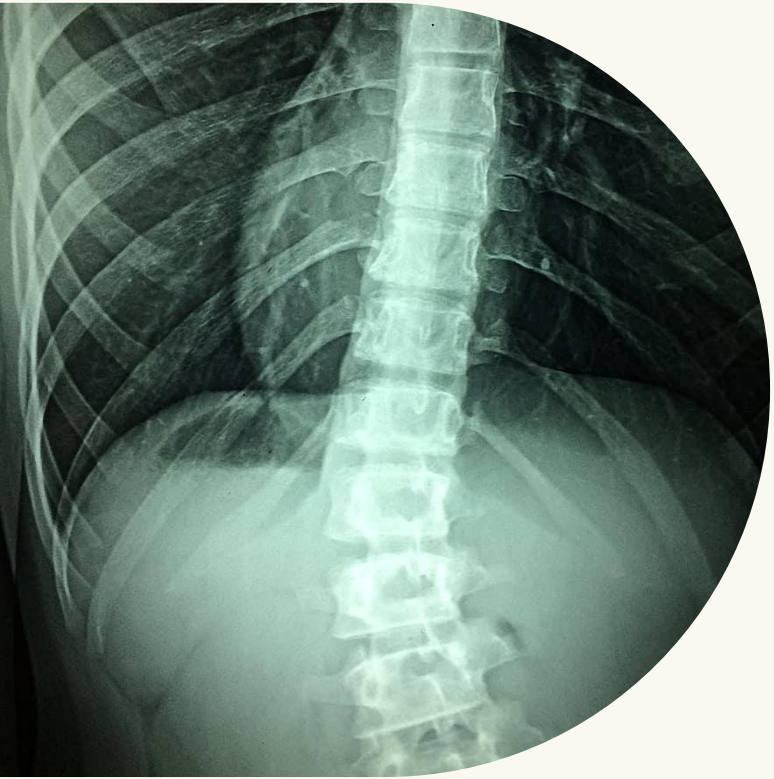
2 categories

(Normal / pneumonia)

Preprocessing

- Canny Edge detection (Not worked for us..edge detection removed the x-ray image details)
- Histogram Equalization (Much improvement gained in F1 score of SVM model)
- Normalization (Get better result in F1 score of SVM model also get better result in CNN models)

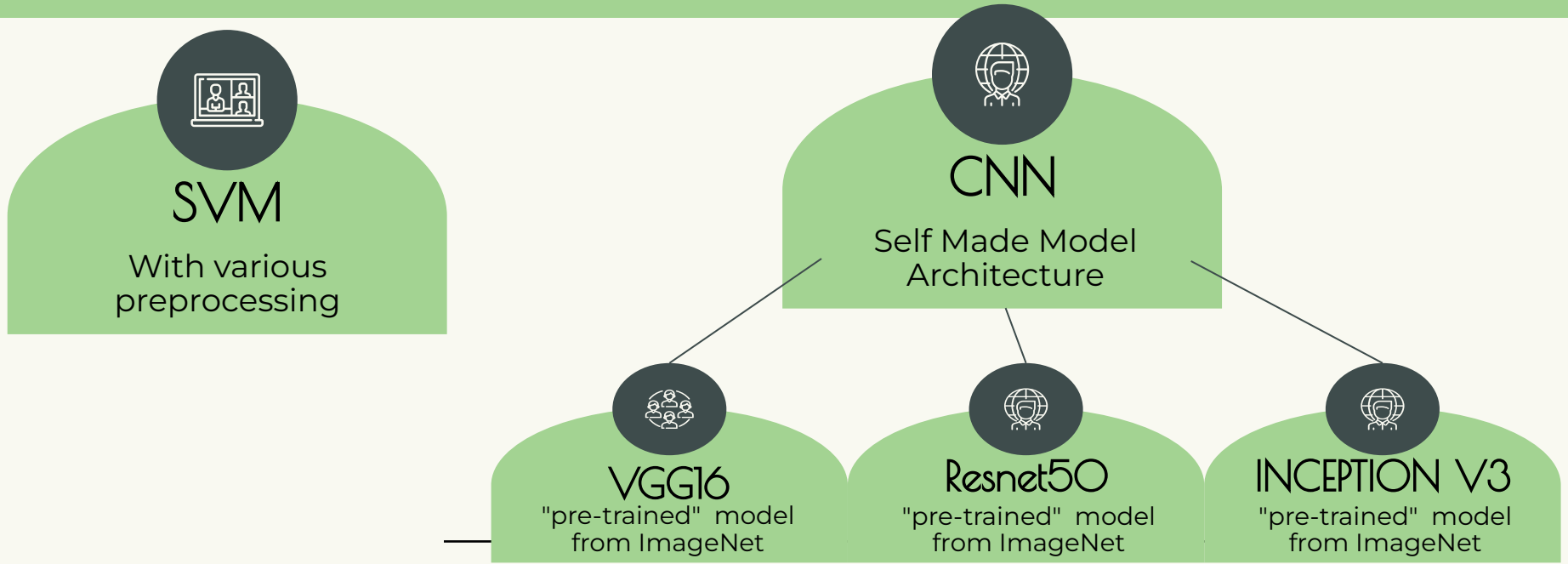
Datas: https://drive.google.com/drive/folders/1AP9QUINzhLtVTip0akIIK44Mb5UgtQ8M?usp=sharing



03

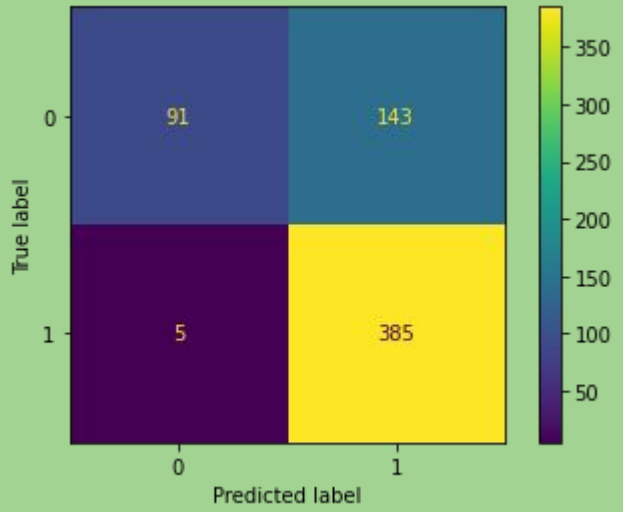
Models

Deep Learning Model

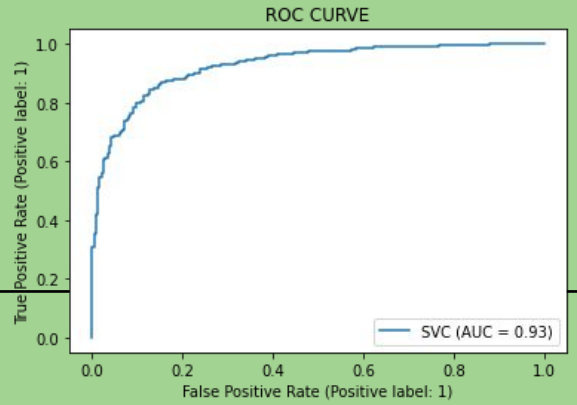


SVM

SVM with normal dataset :
https://colab.research.google.com/drive/1T-Ji_gUBGqLYMrYjhwN4jTFs6VV1pKAj?usp=sharing

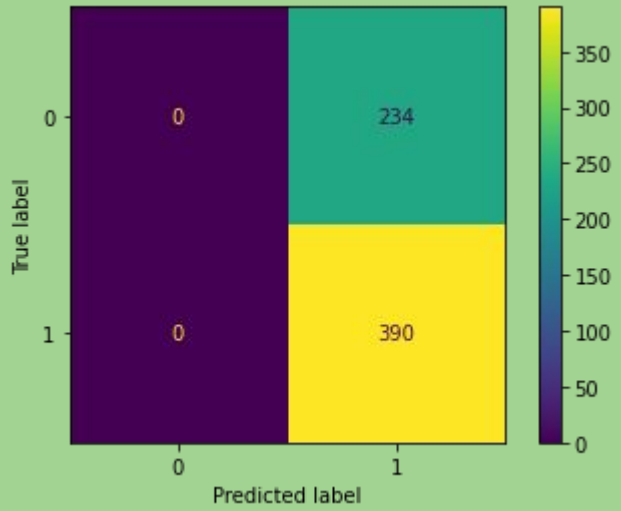


	precision	recall	f1-score	support
0	0.95	0.39	0.55	234
1	0.73	0.99	0.84	390
accuracy			0.76	624
macro avg	0.84	0.69	0.70	624
weighted avg	0.81	0.76	0.73	624

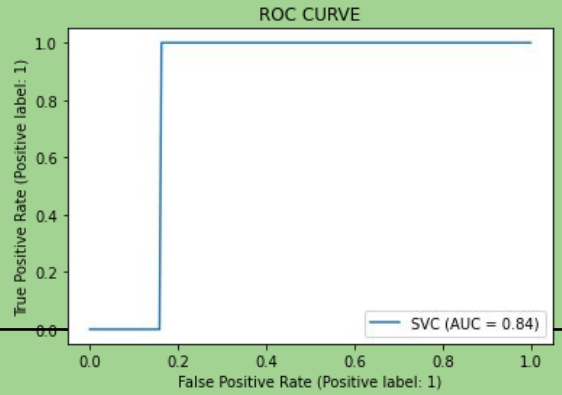


SVM

SVM with Canny Edge processed dataset:
https://colab.research.google.com/drive/119ugD19beoNXHoC17SJhFNmgCCSUsJ6N?usp=sharing

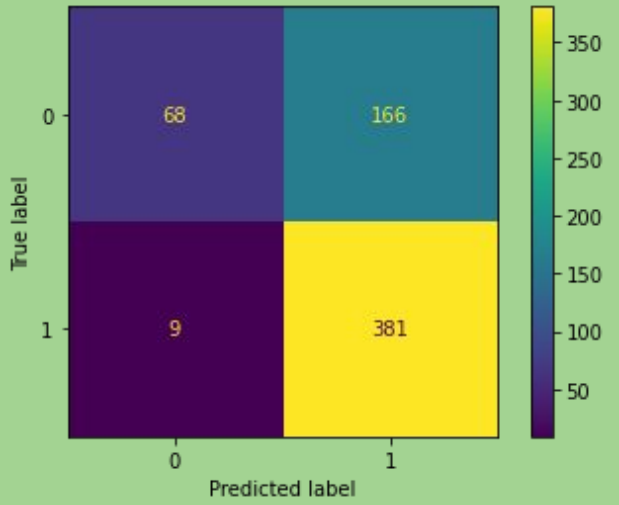


	precision	recall	f1-score	support
0	0.00	0.00	0.00	234
1	0.62	1.00	0.77	390
accuracy			0.62	624
macro avg	0.31	0.50	0.38	624
weighted avg	0.39	0.62	0.48	624

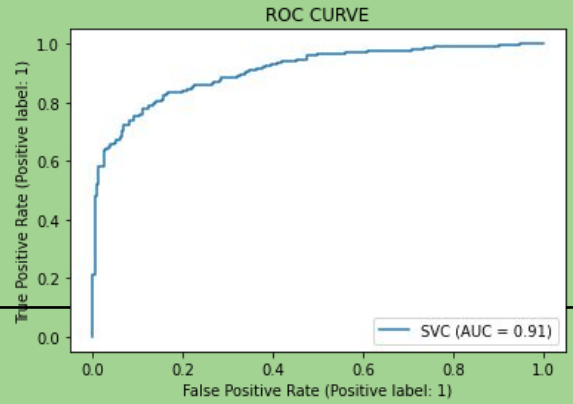


SVM

SVM with Histogram Equalizalized dataset:
https://colab.research.google.com/drive/1LBZcNm-RNn1kSs6Y-9wbydMw6H0X-Ipa?usp=sharing

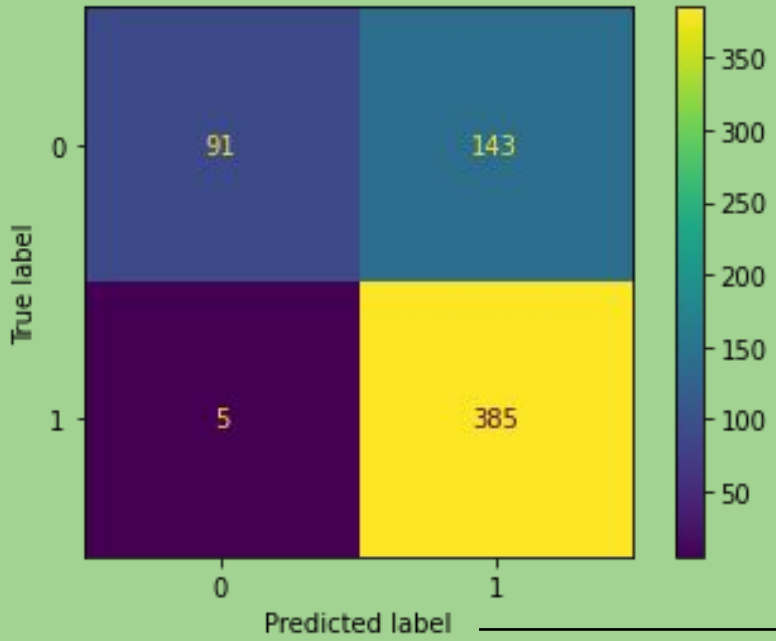


	precision	recall	f1-score	support
0	0.94	0.88	0.91	1341
1	0.96	0.98	0.97	3875
accuracy			0.95	5216
macro avg	0.95	0.93	0.94	5216
weighted avg	0.95	0.95	0.95	5216

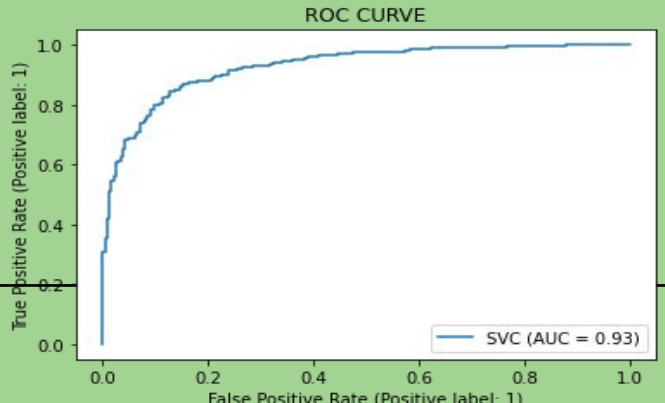


SVM

SVM after normalization of image:
https://colab.research.google.com/drive/1aJBIp0Bt9jS1IKH9Kw1d4GZgh5MKNhLP?usp=sharing



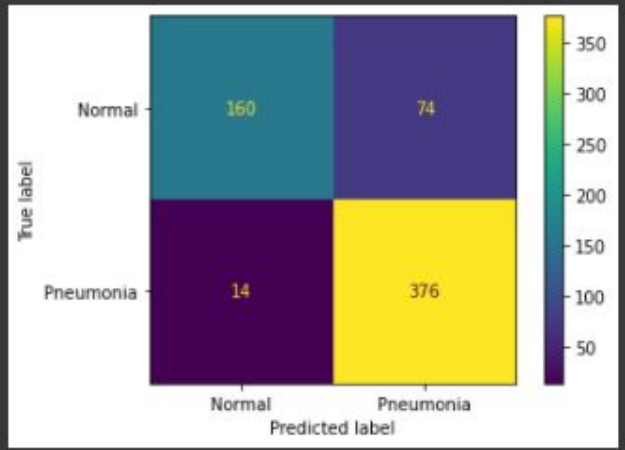
	precision	recall	f1-score	support
0	0.95	0.39	0.55	234
1	0.73	0.99	0.84	390
accuracy			0.76	624
macro avg	0.84	0.69	0.70	624
weighted avg	0.81	0.76	0.73	624



CNN with 15 layers

```
batch_size = 32
target_size = (116,82)

train_datagen = ImageDataGenerator(rescale=1./255.0,
                                   validation_split=0.3,
                                   zoom_range=0.8,
                                   horizontal_flip=True,
                                  )
test_datagen=ImageDataGenerator(rescale = 1./255.0)
```

```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7f0b1fa22e90>
              precision    recall  f1-score   support

      NORMAL       0.92      0.68      0.78       234
   PNEUMONIA       0.84      0.96      0.90       390

    accuracy                           0.86       624
   macro avg       0.88      0.82      0.84       624
weighted avg       0.87      0.86      0.85       624
```

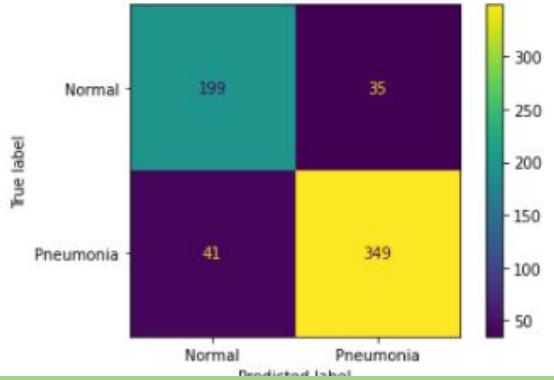


CNN with more than 15 layers

more preprocessing and using AveragePooling layer

```
train_datagen = ImageDataGenerator(rescale=1./255.0,
                                   validation_split=0.3,
                                   rotation_range=40,
                                   width_shift_range=0.2,
                                   height_shift_range=0.2,
                                   zoom_range=0.2,
                                   fill_mode='nearest',
                                  horizontal_flip=True,
                                  )
test_datagen=ImageDataGenerator(rescale = 1./255.0)
```

```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7f46684d6690>
              precision    recall  f1-score   support

      NORMAL       0.83      0.85      0.84       234
   PNEUMONIA       0.91      0.89      0.90       390

    accuracy                           0.88       624
   macro avg       0.87      0.87      0.87       624
weighted avg       0.88      0.88      0.88       624
```

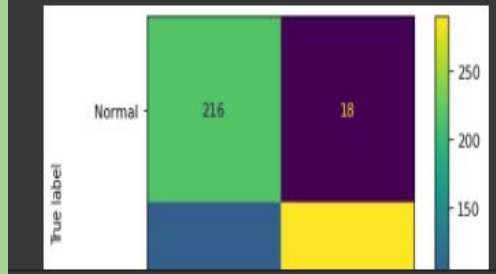


CNN with less than 15 layers

More pre-processing, less strides and layers

```
model.add(layers.Conv2D(64, kernel_size=(3, 3), activation='relu', input_shape=input_shape))
model.add(layers.Conv2D(64, kernel_size=(3, 3), activation='relu', padding='same'))

model.add(layers.MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))
model.add(layers.AveragePooling2D(pool_size=(2, 2), strides=(2, 2)))
model.add(layers.Dropout(0.25))

model.add(layers.Flatten())

model.add(layers.Dense(1024, activation='relu'))
model.add(layers.Dropout(0.5))
#model.add(layers.Dense(1, activation='softmax'))
model.add(layers.Dense(1, activation='sigmoid'))
```

```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7f697ad47
              precision    recall  f1-score   support

      NORMAL       0.68      0.92      0.79       234
   PNEUMONIA       0.94      0.74      0.83       390

    accuracy                           0.81       624
   macro avg       0.81      0.83      0.81       624
weighted avg       0.84      0.81      0.81       624
```



CNN (*Self Made Model Architecture*)

CNN with 15 layers
https://colab.research.google.com/drive/1sjzbnPUW2AWlv26EwBgMra3s-W_GJSTC?usp=sharing

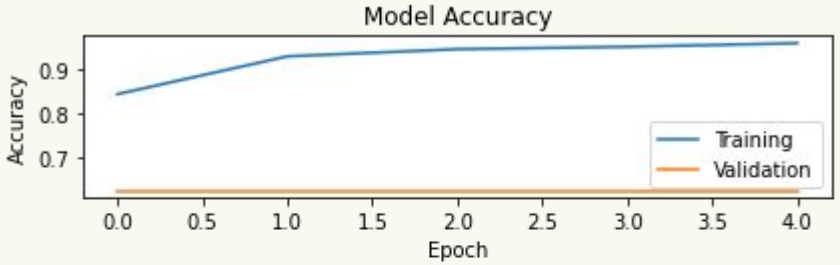
CNN with greater layer and more pre-processing
https://colab.research.google.com/drive/1Ue7L-0OtyhxTg_qqjKWSOMsi01lXHBu-?usp=sharing

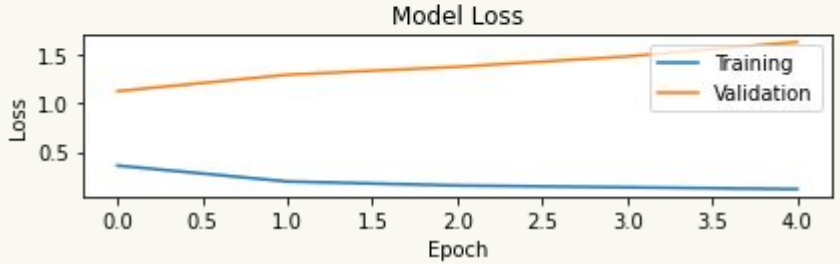
CNN with 6 layers
https://colab.research.google.com/drive/1YL-YSFkGPUOU6nKPwjTGWeEmkNMnyUbS?usp=sharing

Resnet50

Preprocessing:

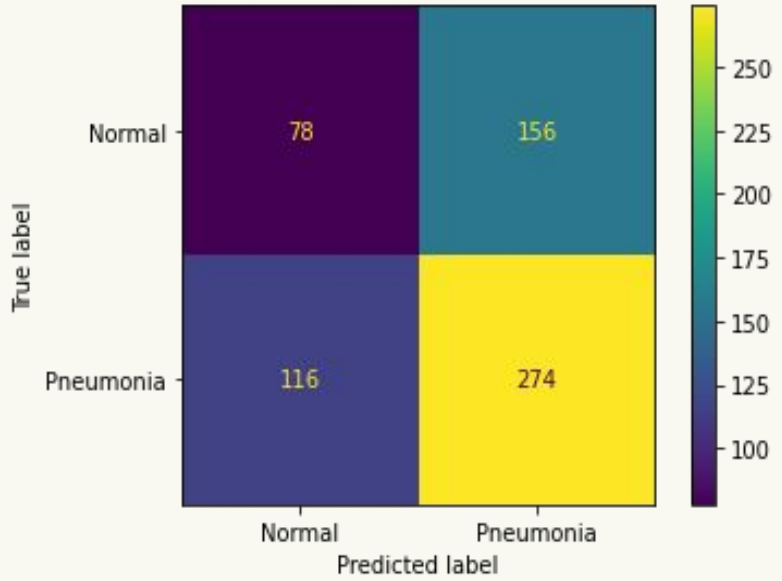
```
[ ]  train_datagen = ImageDataGenerator(
         rescale=1. / 255,
         shear_range=0.2,
         zoom_range=0.2,
         horizontal_flip=True)
```





VGG16

NORMAL	0.40	0.33	0.36	234
PNEUMONIA	0.64	0.70	0.67	390
accuracy			0.56	624
macro avg	0.52	0.52	0.52	624
weighted avg	0.55	0.56	0.55	624



Preprocessing:

```
[ ] train_datagen = ImageDataGenerator(
        rescale=1. / 255,
        shear_range=0.2,
        zoom_range=0.2,
        horizontal_flip=True)
```

Challenges

- Resnet50 and VGG16 took a lot of time to run.
- Before saving the model of Resnet50, runtime ended.
- VGG16 model was created with only 10 epochs.

Resnet50

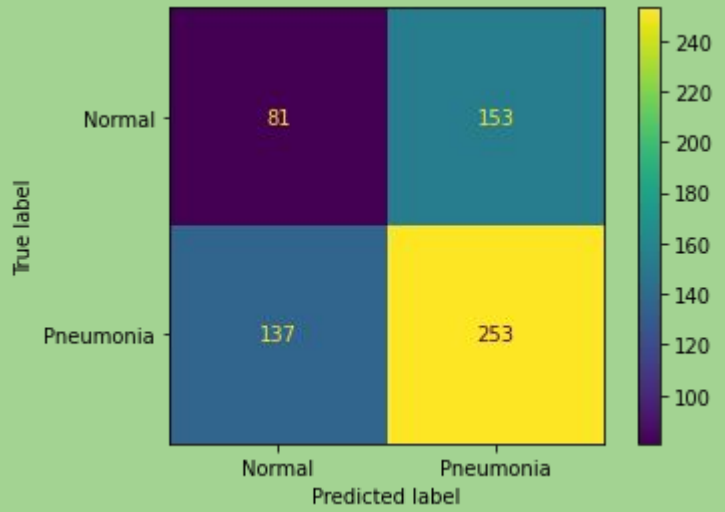
Resnet50:
https://colab.research.google.com/drive/1VTLkXAoKwLys2wUawhxxOgW_e1B-xaM0?usp=sharing

VGG16

VGG16:
https://colab.research.google.com/drive/1VTLkXAoKwLys2wUawhxxOgW_e1B-xaM0?usp=sharing

Inception V3

InceptionV3 with normal dataset :
https://colab.research.google.com/drive/1OseEMHs2pLeYb3rIJQFNGvhxCXlqfJdW?usp=sharing



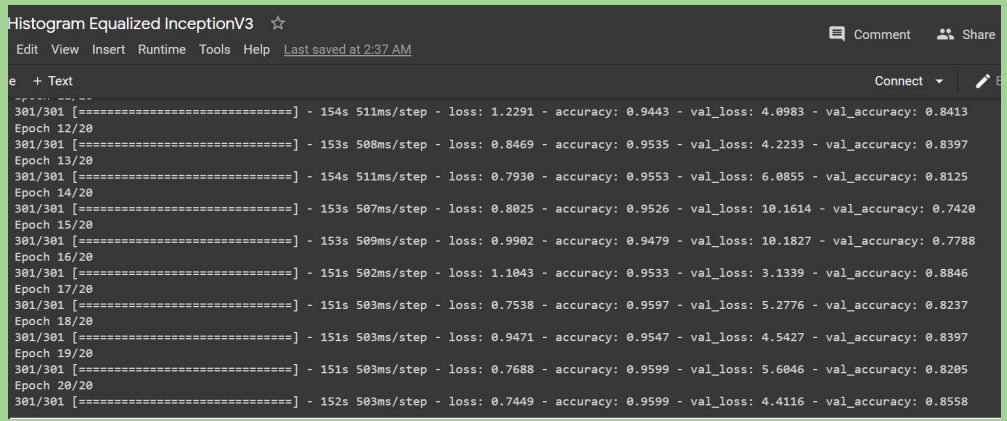
Data Augmentation

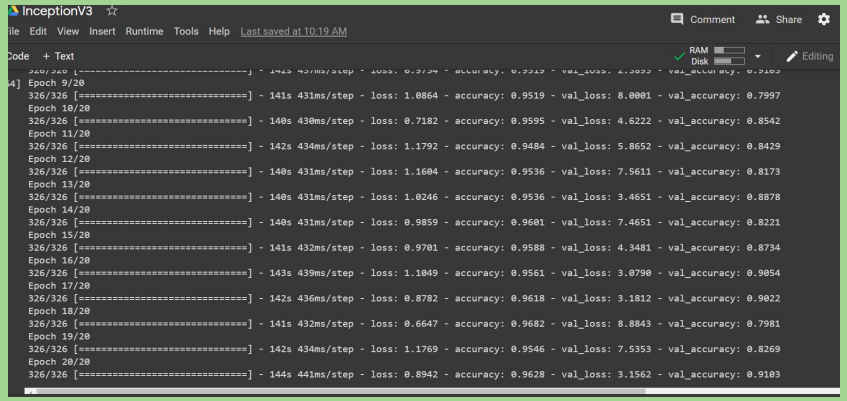
```
train_datagen = ImageDataGenerator(rescale=1./255,
                                   shear_range=0.2,
                                   zoom_range=0.2,
                                   horizontal_flip=True)
test_datagen = ImageDataGenerator(rescale = 1./255)
```

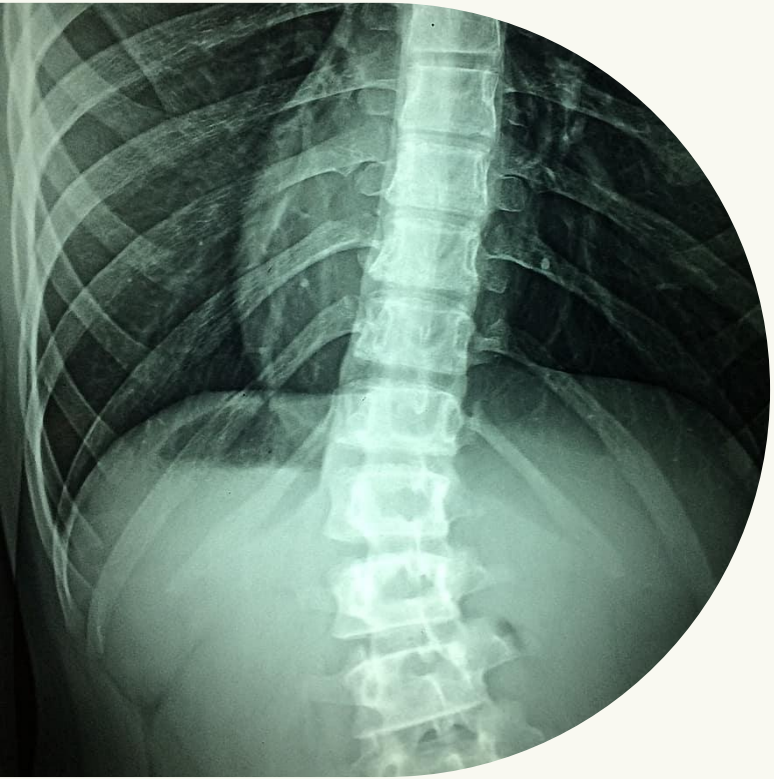
	precision	recall	f1-score	support
NORMAL	0.37	0.35	0.36	234
PNEUMONIA	0.62	0.65	0.64	390
accuracy			0.54	624
macro avg	0.50	0.50	0.50	624
weighted avg	0.53	0.54	0.53	624

Inception V3

InceptionV3 with Histogram Equalizalized dataset:
https://colab.research.google.com/drive/1q6CCi_fIXLuJxfVZ1giY5pLhx0Cdd98a?usp=sharing







04

Evaluation Techniques

Evaluation Techniques

ACCURACY

TP+TN / TP+TN+FP+FN

PRECISIÓN

TP / (TP + FP)

RECALL

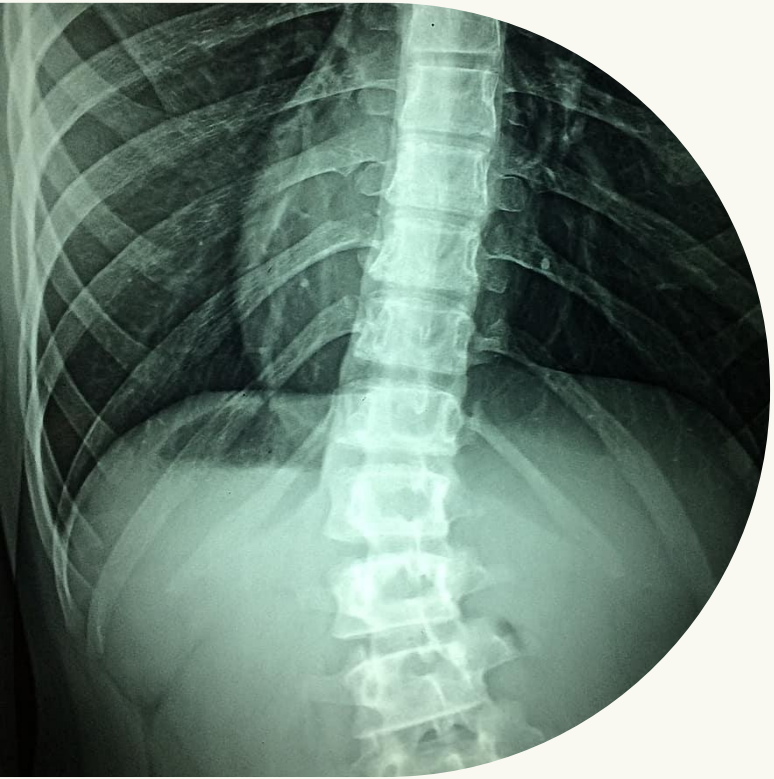
TP / (TP + FN)
most Important for this particular problem

F1 SCORE

(TP + TN) / Total

Comparison

MODEL	PREPROCESSING	Accuracy	Precisión	Recall	**F1 Score**
SVM	Resizing	0.76	0.73	0.99	0.84
	Canny edge	0.62	0.62	1.0	0.77
	Histogram Eq.	0.95	0.96	0.98	0.97
	Normalization	0.76	0.73	0.99	0.84
CNN	rescale, zoom, horizontal flip (15)	0.86	0.84	0.96	0.90
	rotation, shift (>15)	0.88	0.91	0.89	0.90
	same (<15)	0.81	0.94	0.74	0.83
InceptionV3		0.54	0.62	0.65	0.64
VGG16		0.56	0.64	0.70	0.67



05

Limitations

Limitations

- We couldn’t save the best model of InceptionV3. (should’ve used callback)
- Couldn’t compare the evaluation metric of Inception V3 with histogram eq. and Resnet50.

THANKS!