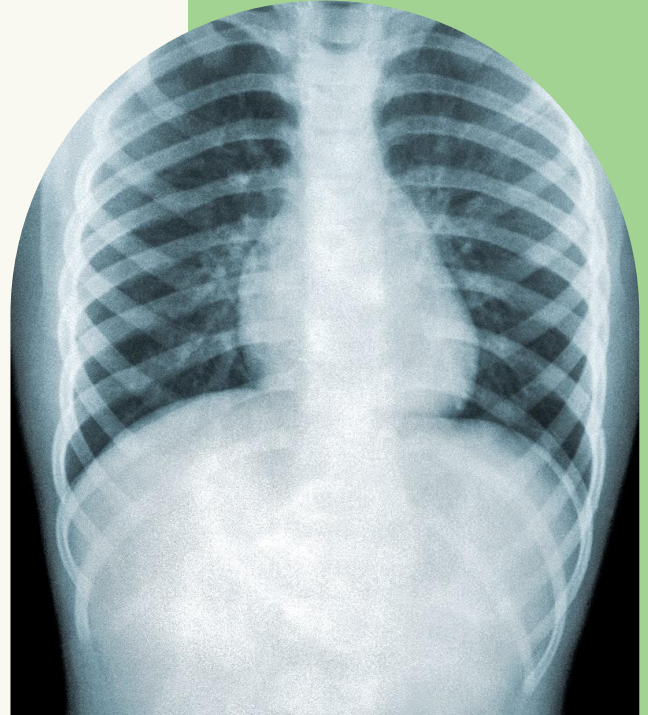


Pneumonia Detection Using Chest X-Ray Image

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

O5

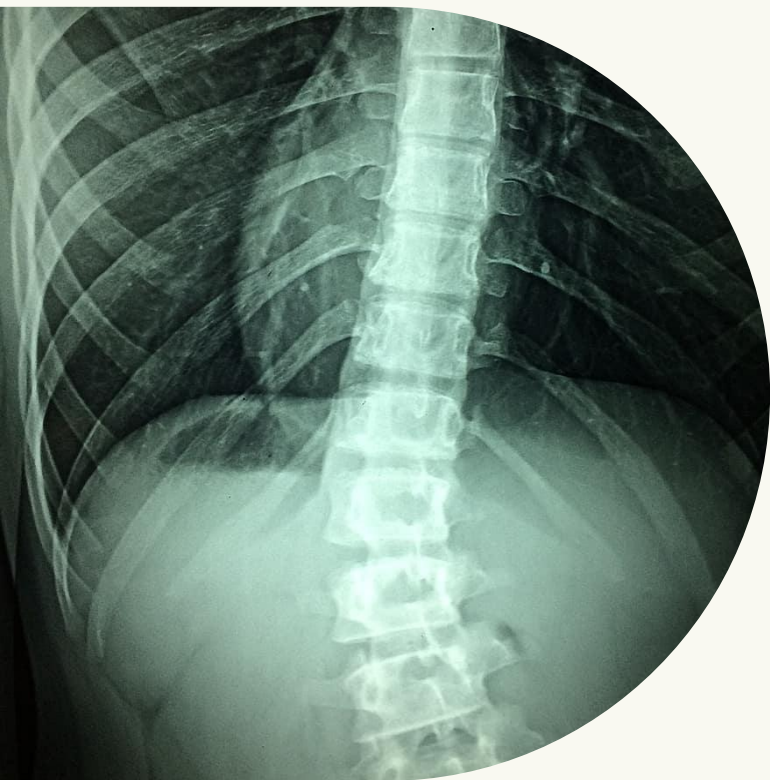
Evaluation

O6

Limitations

O7

Running the model on a
data point



Problem Definition and Proposed Solution



Problem Definition and Proposed Solution

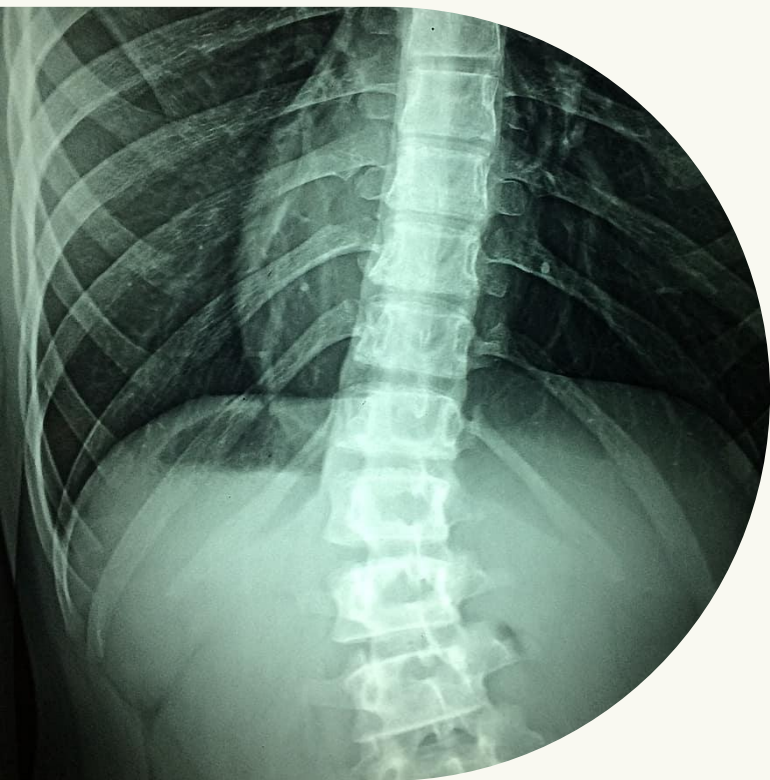
- Pneumonia is one of the major illnesses in children and aged humans due to the Infection in the lungs.
 - Analysis of pneumonia is necessary to prepare for a possible treatment procedure to regulate and cure the disease.
 - Chest X-ray imaging is the most frequently used method for diagnosing pneumonia.
 - 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.





O2

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)



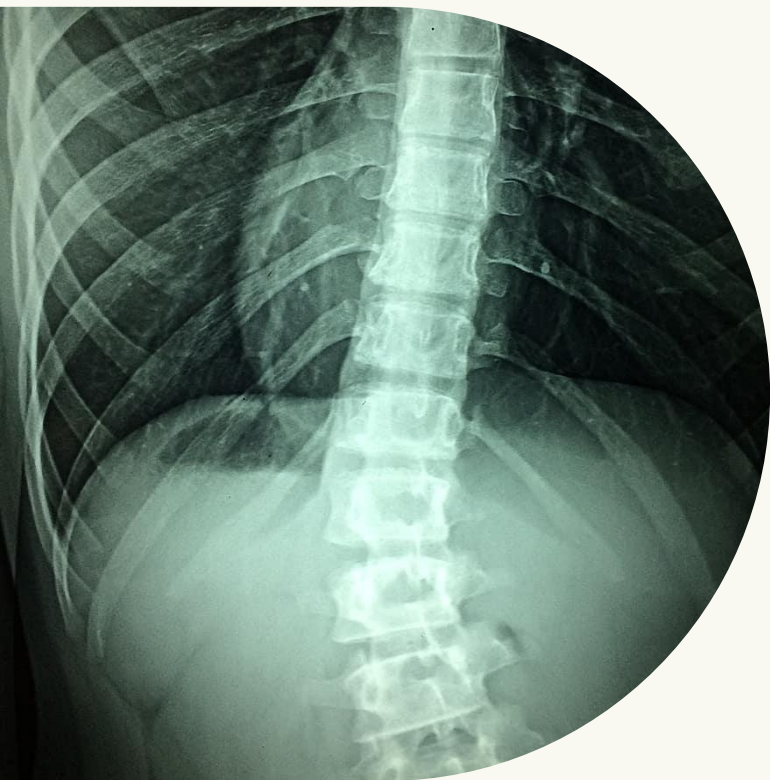
2 categories

(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)

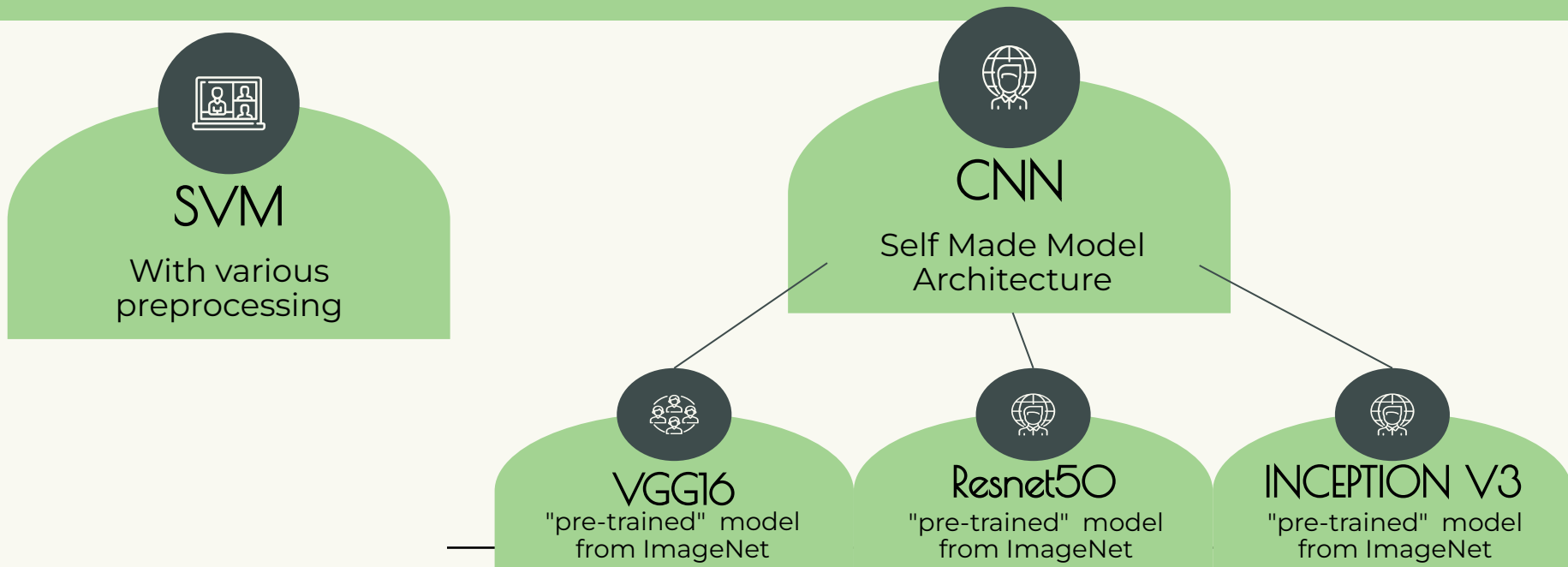
Datas: <https://drive.google.com/drive/folders/1AP9QUINzhLtVTip0akIK44Mb5UqtQ8M?usp=sharing>



O3

Models

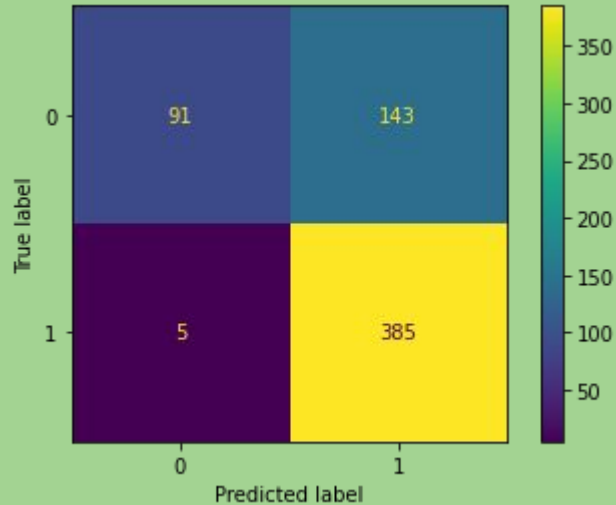
Deep Learning Model



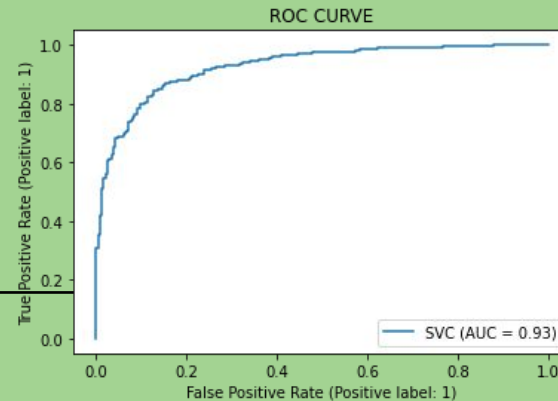
SVM

SVM with normal dataset :

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



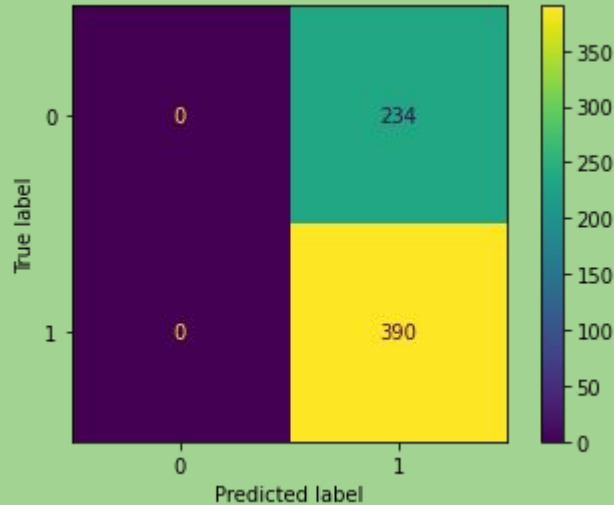
	precision	recall	f1-score	support
0	0.95	0.39	0.55	234
1	0.73	0.99	0.84	390
accuracy			0.76	624
macro avg	0.84	0.69	0.70	624
weighted avg	0.81	0.76	0.73	624



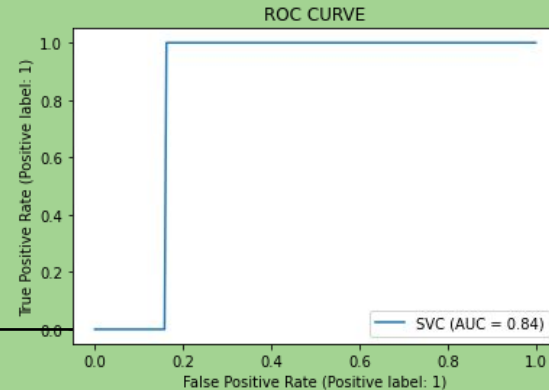
SVM

SVM with Canny Edge processed dataset:

<https://colab.research.google.com/drive/119ugD19beoNXHoC17SJhFNmgCCSUsJ6N?usp=sharing>



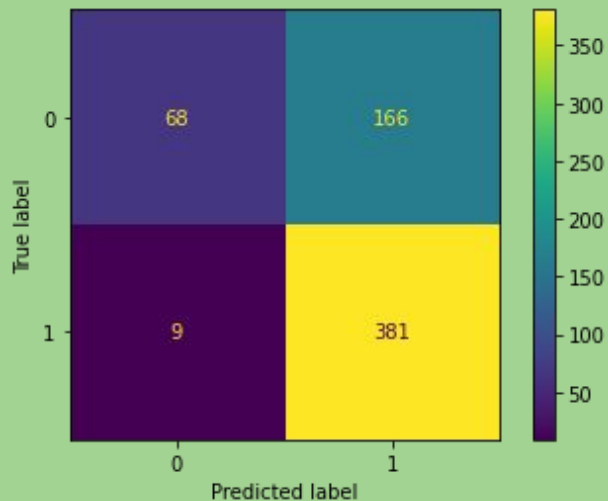
	precision	recall	f1-score	support
0	0.00	0.00	0.00	234
1	0.62	1.00	0.77	390
accuracy			0.62	624
macro avg	0.31	0.50	0.38	624
weighted avg	0.39	0.62	0.48	624



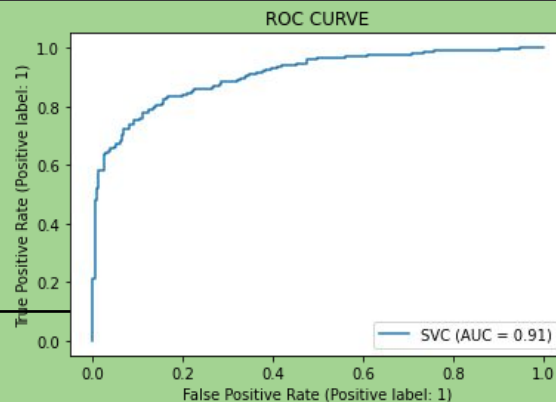
SVM

SVM with Histogram Equalized dataset:

<https://colab.research.google.com/drive/1LBZcNm-RNn1kSs6Y-9wbydMw6H0X-lpa?usp=sharing>



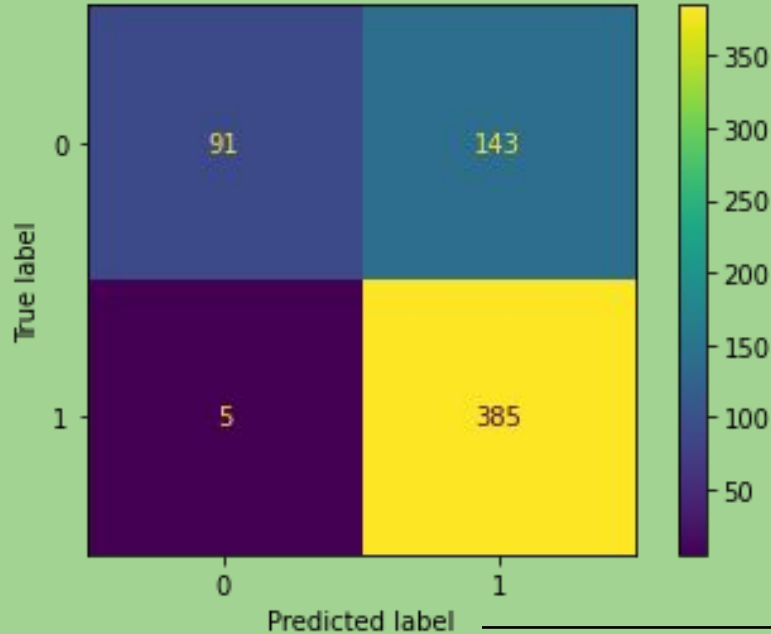
	precision	recall	f1-score	support
0	0.94	0.88	0.91	1341
1	0.96	0.98	0.97	3875
accuracy			0.95	5216
macro avg	0.95	0.93	0.94	5216
weighted avg	0.95	0.95	0.95	5216



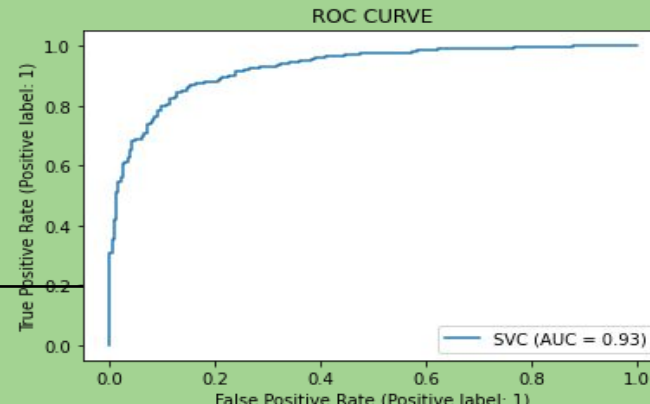
SVM

SVM after normalization of image:

<https://colab.research.google.com/drive/1aJB1p0Bt9jS1IKH9Kw1d4GZgh5MKNhLP?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

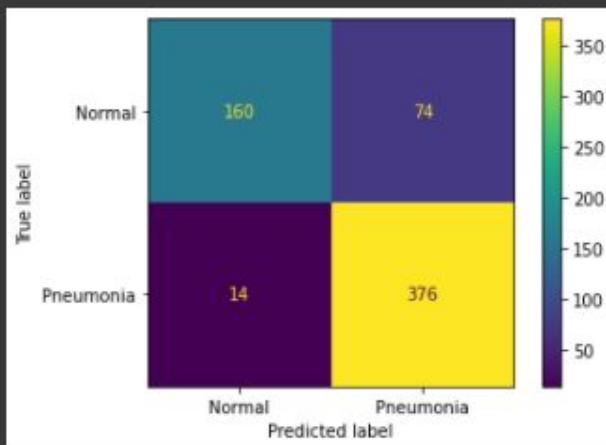
```
batch_size = 32  
target_size = (116,82)
```

```
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



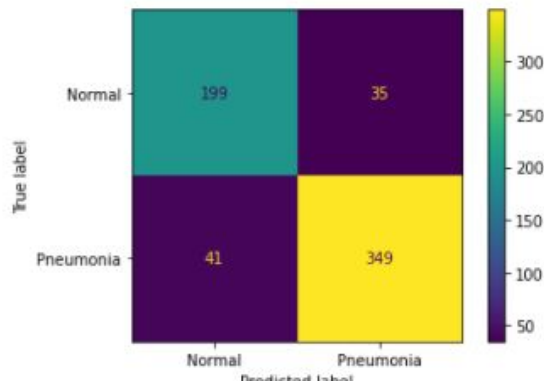
CNN with more than 15 layers

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

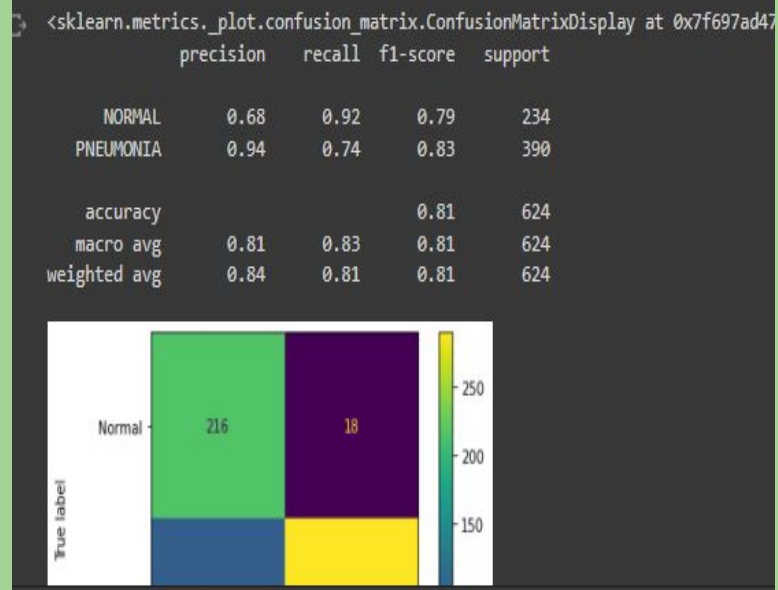
More pre-processing, less strides and layers

```
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'))
```



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_qqjKWSOMsi01IXHBu-?usp=sharing

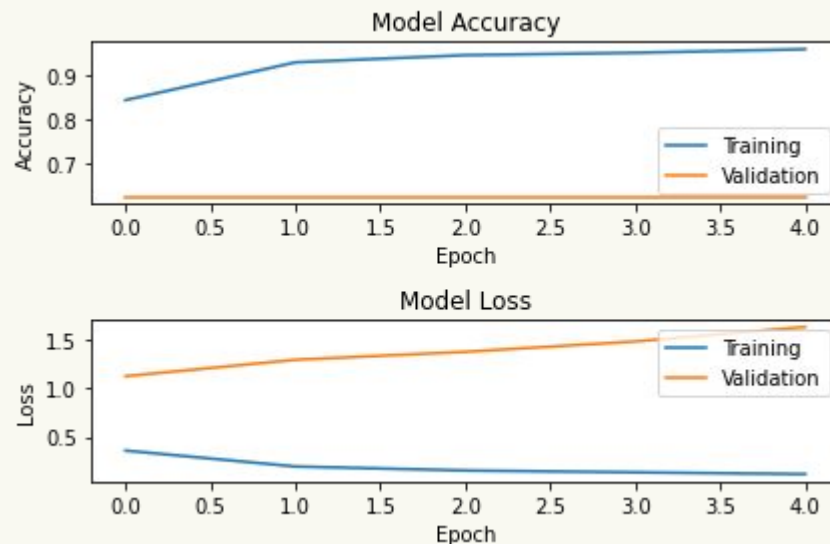
CNN with 6 layers

<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)
```

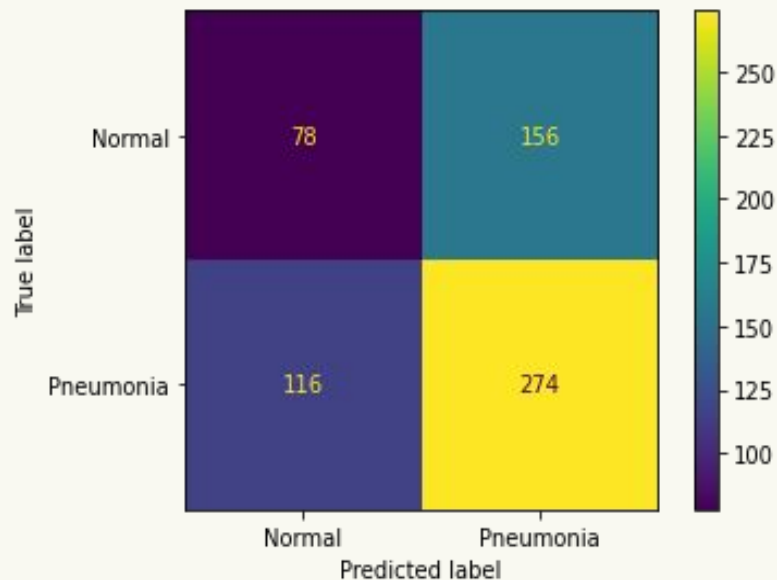


VGG16

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:

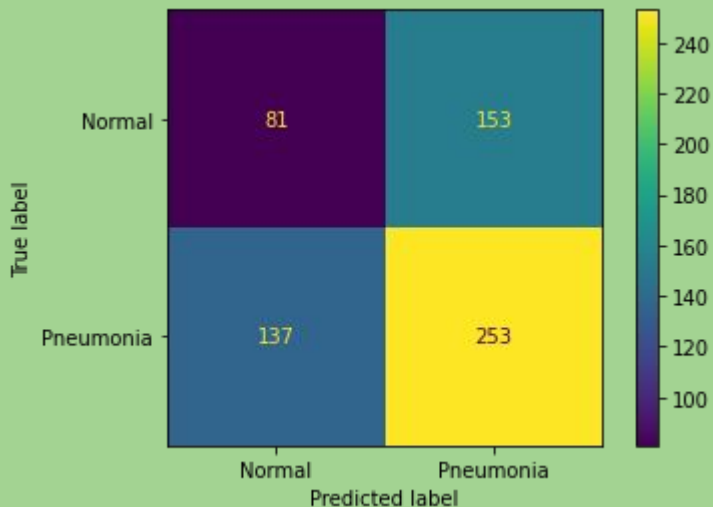
https://colab.research.google.com/drive/1VTLkXAoKwLys2wUawhxxOgW_e1B-xaM0?usp=sharing

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 Equalized dataset:

https://colab.research.google.com/drive/1d6CCi_fiXLuJxfVZlqiY5pLxh0Cdd98a?usp=sharing

```
Histogram Equalized InceptionV3 ☆
Edit View Insert Runtime Tools Help Last saved at 2:37 AM

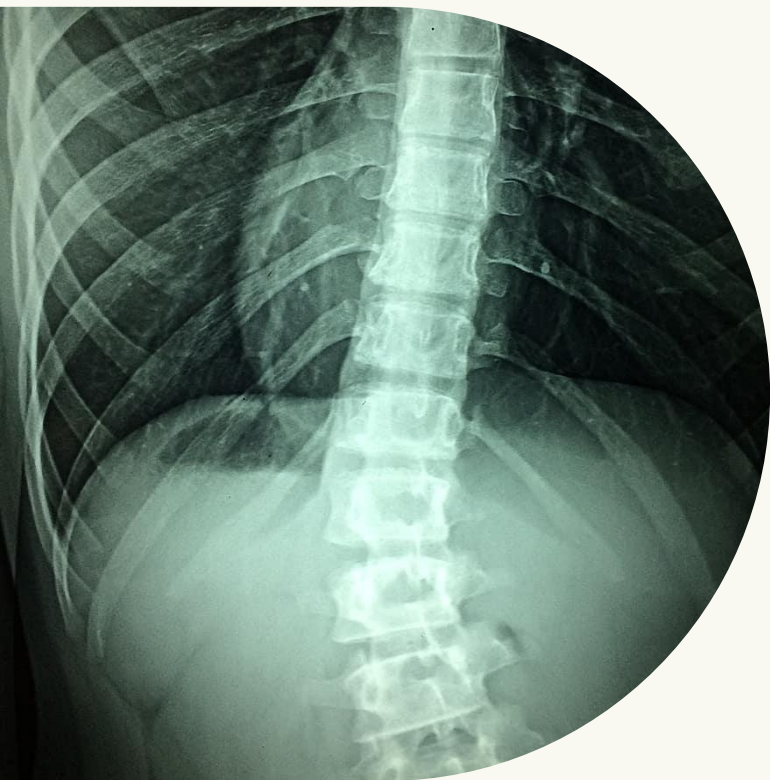
e + Text Connect

301/301 [=====] - 154s 511ms/step - loss: 1.2291 - accuracy: 0.9443 - val_loss: 4.0983 - val_accuracy: 0.8413
Epoch 12/20
301/301 [=====] - 153s 508ms/step - loss: 0.8469 - accuracy: 0.9535 - val_loss: 4.2233 - val_accuracy: 0.8397
Epoch 13/20
301/301 [=====] - 154s 511ms/step - loss: 0.7930 - accuracy: 0.9553 - val_loss: 6.0855 - val_accuracy: 0.8125
Epoch 14/20
301/301 [=====] - 153s 507ms/step - loss: 0.8025 - accuracy: 0.9526 - val_loss: 10.1614 - val_accuracy: 0.7420
Epoch 15/20
301/301 [=====] - 153s 509ms/step - loss: 0.9902 - accuracy: 0.9479 - val_loss: 10.1827 - val_accuracy: 0.7788
Epoch 16/20
301/301 [=====] - 151s 502ms/step - loss: 1.1043 - accuracy: 0.9533 - val_loss: 3.1339 - val_accuracy: 0.8846
Epoch 17/20
301/301 [=====] - 151s 503ms/step - loss: 0.7538 - accuracy: 0.9597 - val_loss: 5.2776 - val_accuracy: 0.8237
Epoch 18/20
301/301 [=====] - 151s 503ms/step - loss: 0.9471 - accuracy: 0.9547 - val_loss: 4.5427 - val_accuracy: 0.8397
Epoch 19/20
301/301 [=====] - 151s 503ms/step - loss: 0.7688 - accuracy: 0.9599 - val_loss: 5.6046 - val_accuracy: 0.8205
Epoch 20/20
301/301 [=====] - 152s 503ms/step - loss: 0.7449 - accuracy: 0.9599 - val_loss: 4.4116 - val_accuracy: 0.8558
```

```
InceptionV3 ☆
File Edit View Insert Runtime Tools Help Last saved at 10:19 AM

Code + Text RAM Disk Editing

301/301 [=====] - 144s 431ms/step - loss: 0.7124 - accuracy: 0.9549 - val_loss: 4.2022 - val_accuracy: 0.7400
Epoch 9/20
326/326 [=====] - 141s 431ms/step - loss: 1.0864 - accuracy: 0.9519 - val_loss: 8.0001 - val_accuracy: 0.7997
Epoch 10/20
326/326 [=====] - 140s 430ms/step - loss: 0.7182 - accuracy: 0.9595 - val_loss: 4.6222 - val_accuracy: 0.8542
Epoch 11/20
326/326 [=====] - 142s 434ms/step - loss: 1.1792 - accuracy: 0.9484 - val_loss: 5.8652 - val_accuracy: 0.8429
Epoch 12/20
326/326 [=====] - 140s 431ms/step - loss: 1.1604 - accuracy: 0.9536 - val_loss: 7.5611 - val_accuracy: 0.8173
Epoch 13/20
326/326 [=====] - 140s 431ms/step - loss: 1.0246 - accuracy: 0.9536 - val_loss: 3.4651 - val_accuracy: 0.8878
Epoch 14/20
326/326 [=====] - 140s 431ms/step - loss: 0.9859 - accuracy: 0.9601 - val_loss: 7.4651 - val_accuracy: 0.8221
Epoch 15/20
326/326 [=====] - 141s 432ms/step - loss: 0.9701 - accuracy: 0.9588 - val_loss: 4.3481 - val_accuracy: 0.8734
Epoch 16/20
326/326 [=====] - 143s 439ms/step - loss: 1.1049 - accuracy: 0.9561 - val_loss: 3.8790 - val_accuracy: 0.9054
Epoch 17/20
326/326 [=====] - 142s 436ms/step - loss: 0.8782 - accuracy: 0.9618 - val_loss: 3.1812 - val_accuracy: 0.9022
Epoch 18/20
326/326 [=====] - 141s 432ms/step - loss: 0.6647 - accuracy: 0.9682 - val_loss: 8.8843 - val_accuracy: 0.7981
Epoch 19/20
326/326 [=====] - 142s 434ms/step - loss: 1.1769 - accuracy: 0.9546 - val_loss: 7.5353 - val_accuracy: 0.8269
Epoch 20/20
326/326 [=====] - 144s 441ms/step - loss: 0.8942 - accuracy: 0.9628 - val_loss: 3.1562 - val_accuracy: 0.9103
```



O4

Evaluation Techniques

Evaluation Techniques



ACCURACY

$$TP+TN / TP+TN+FP+FN$$



RECALL

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



PRECISIÓN

$$TP / (TP + FP)$$

F1 SCORE

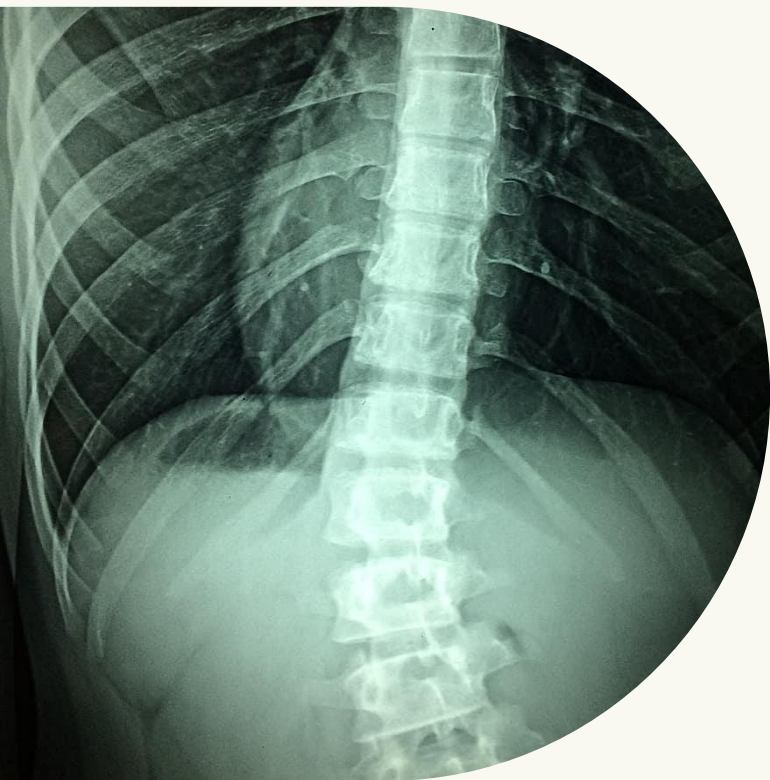
$$(TP + TN) / \text{Total}$$



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



O5

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!