DIAGNOSING

# PNEUMONIA

USING

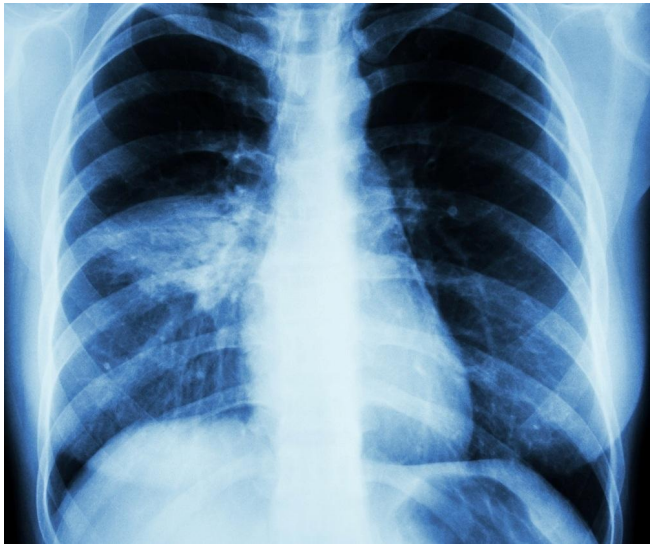
# DEEP LEARNING

Roshna Raj | Juhi Dhakhane | Vidya P
Akash Ghanghav | Anitya Umare | Rahul Raut | Sanket Umap

# Overview

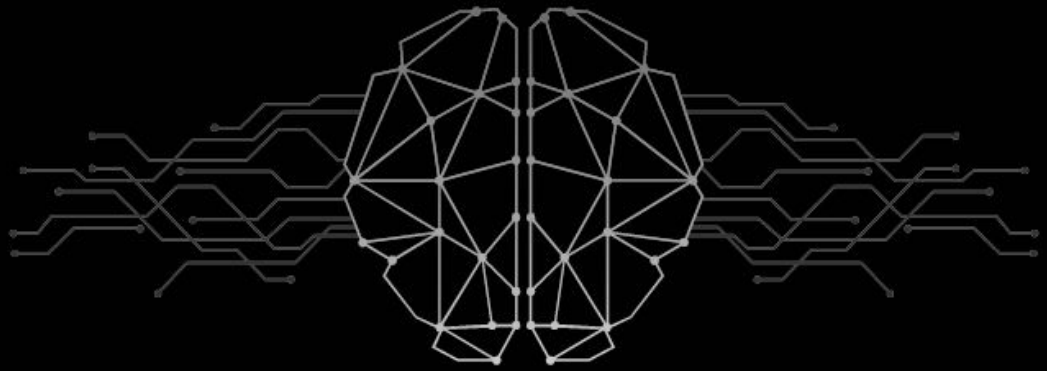
- Introduction
- Deep Learning
- Flow Chart
- Analysis
- Discussion
- References

# Introduction



- **WHO** estimates that over **150 million** people get infected with **Pneumonia** especially children and this is highly prevalent in developing countries.
- In such regions there is dearth of proper medical resources and personnel.
- Hence, **accurate and fast diagnosis** means everything for these populations.
- **Artificial Intelligence** has the potential to revolutionize disease diagnosis by performing classification.

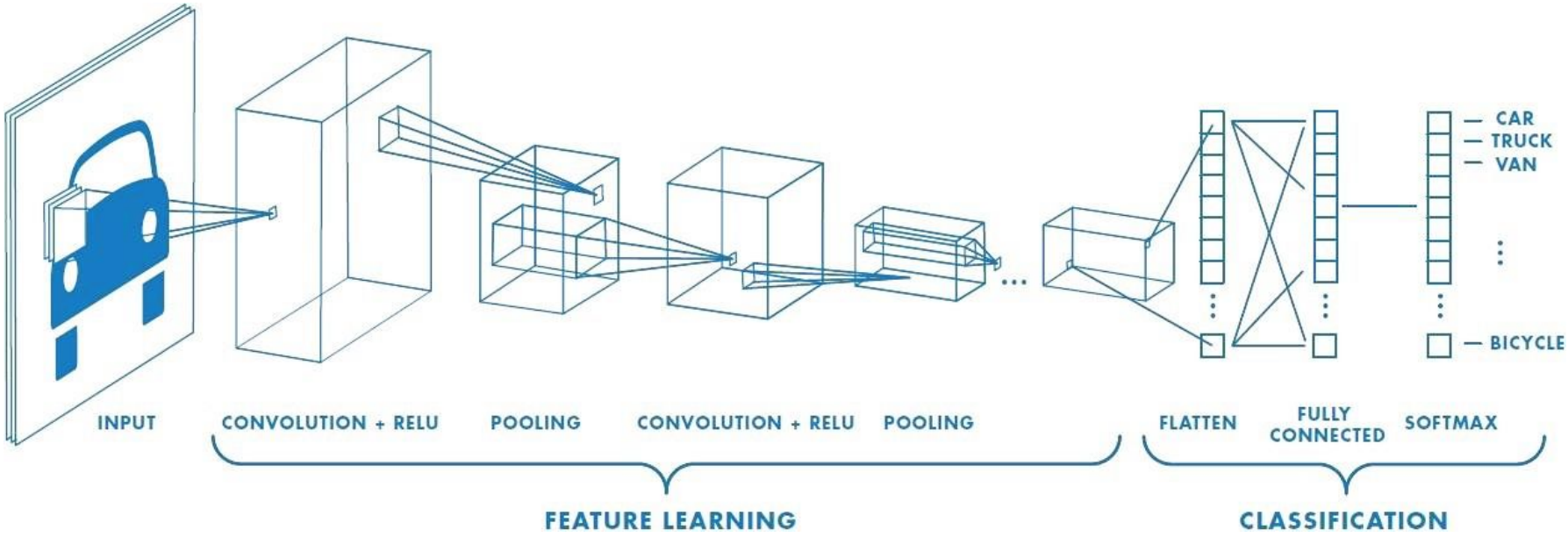
# DEEP LEARNING



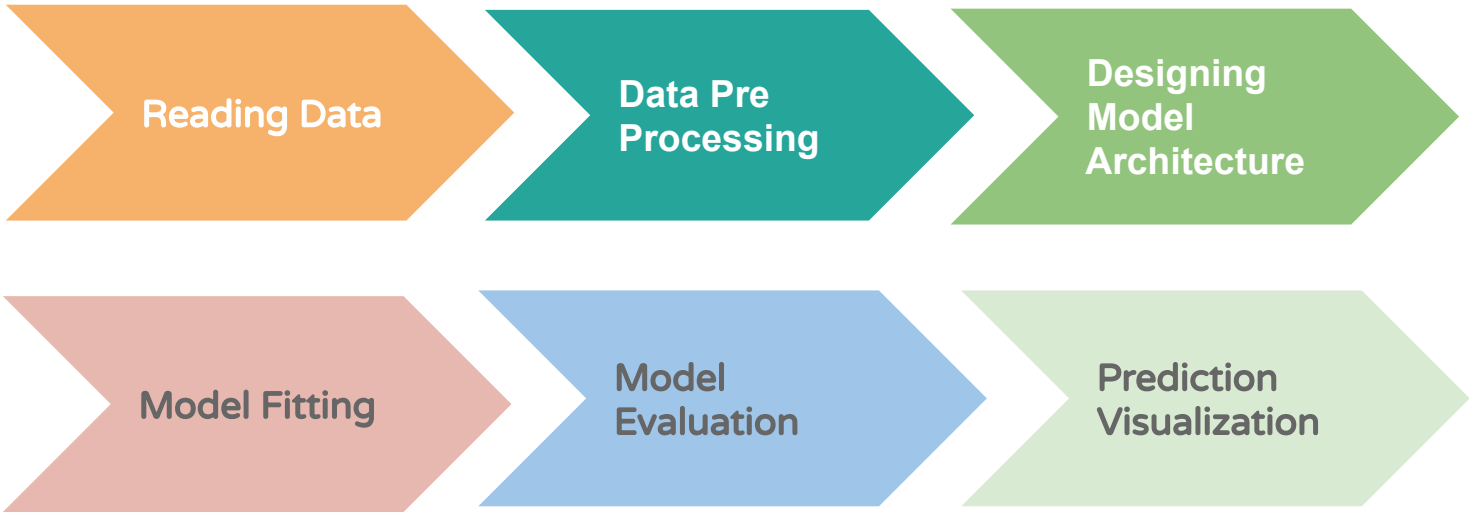
Deep learning is an AI function that imitates the working of the human brain.

Deep learning networks like Convolutional Neural Networks & Recurrent Neural Networks are now widely used for Image & Audio classifications.

# Convolutional Neural Network



# Flow Chart


Reading Data
Data Pre Processing
Designing Model Architecture
Model Fitting
Model Evaluation
Prediction Visualization

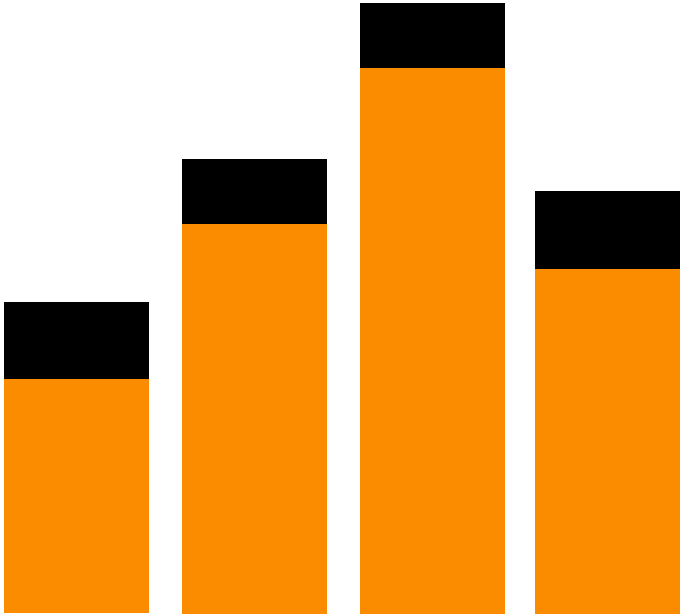
# Analysis

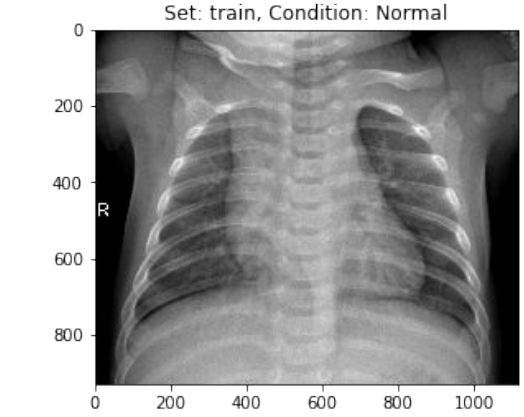
## Method I - Using Keras

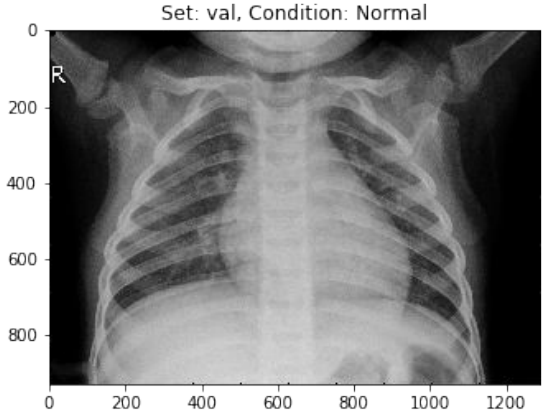
Keras is a high-level neural network API capable of running on Tensorflow, Theano and CNTK.

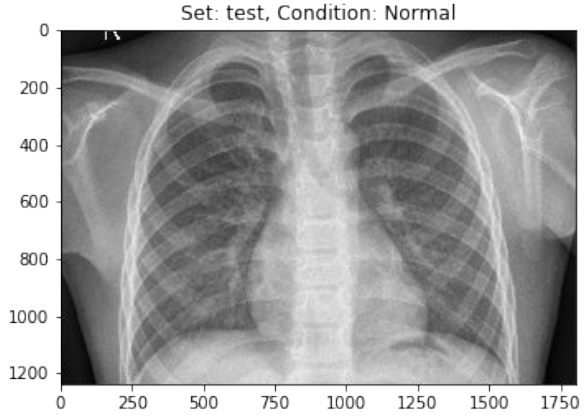
**Steps :**

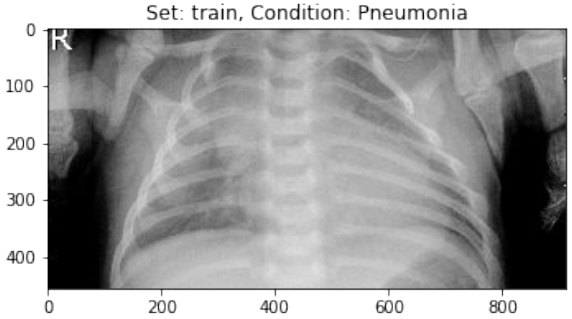
- Import the necessary libraries
- Reading image data and visualizing it
- Image data preprocessing
- Setting the hyper parameters
- Designing Model Architecture
- Fitting model and evaluating it
- Prediction and visualization

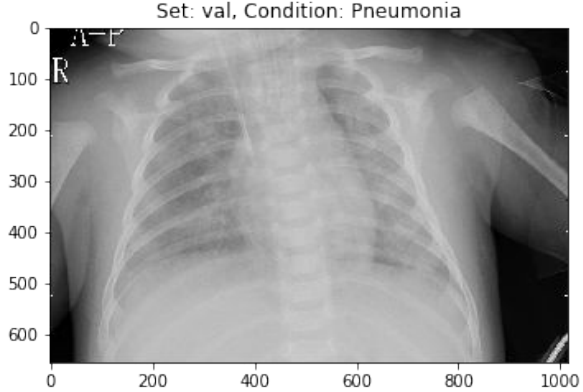


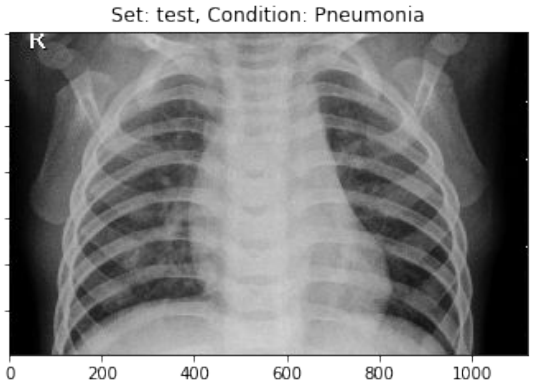



```
Set: train, normal images: 1342, pneumonia images: 3876
Set: val, normal images: 9, pneumonia images: 9
Set: test, normal images: 234, pneumonia images: 390
```

Convolution Layer



Pooling Layer

Separable Convolution Layer



Batch Normalization Layer



Dropout Layer

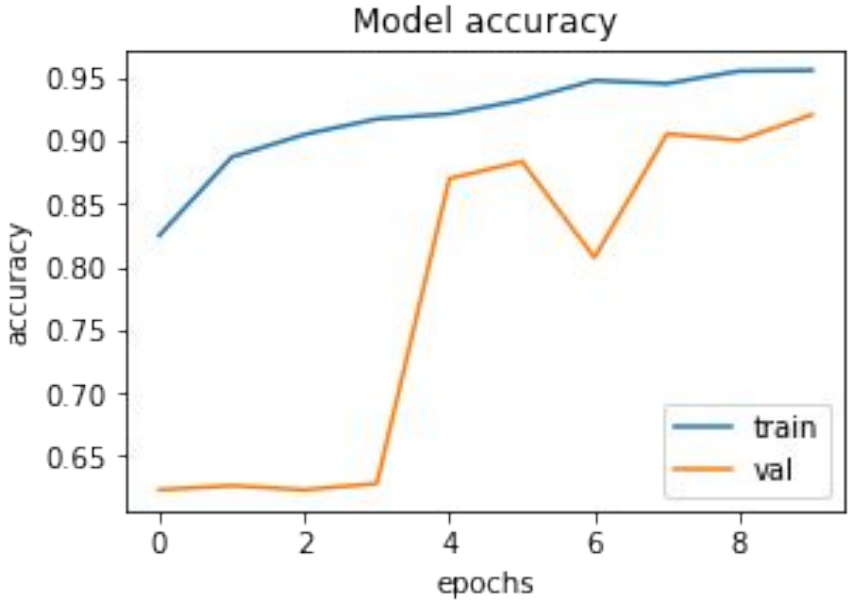
Flatten Layer

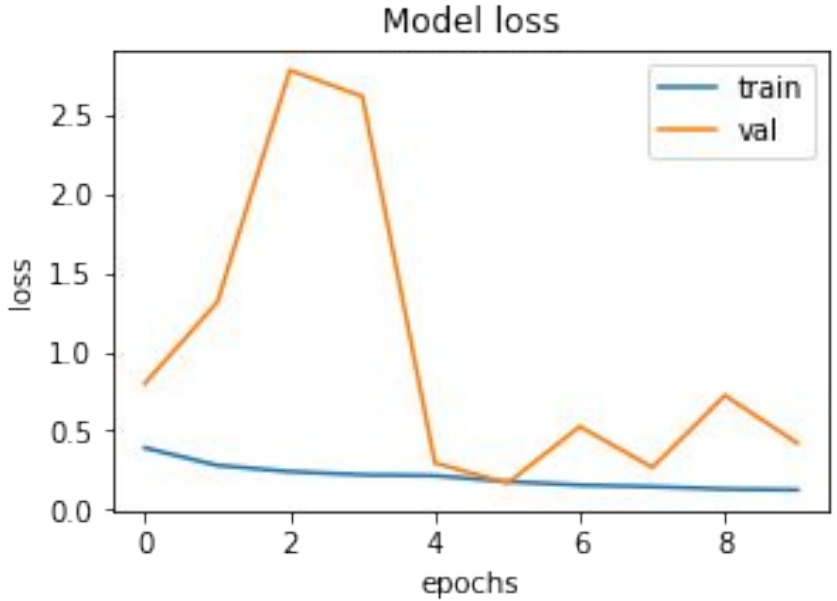


Dense Layer

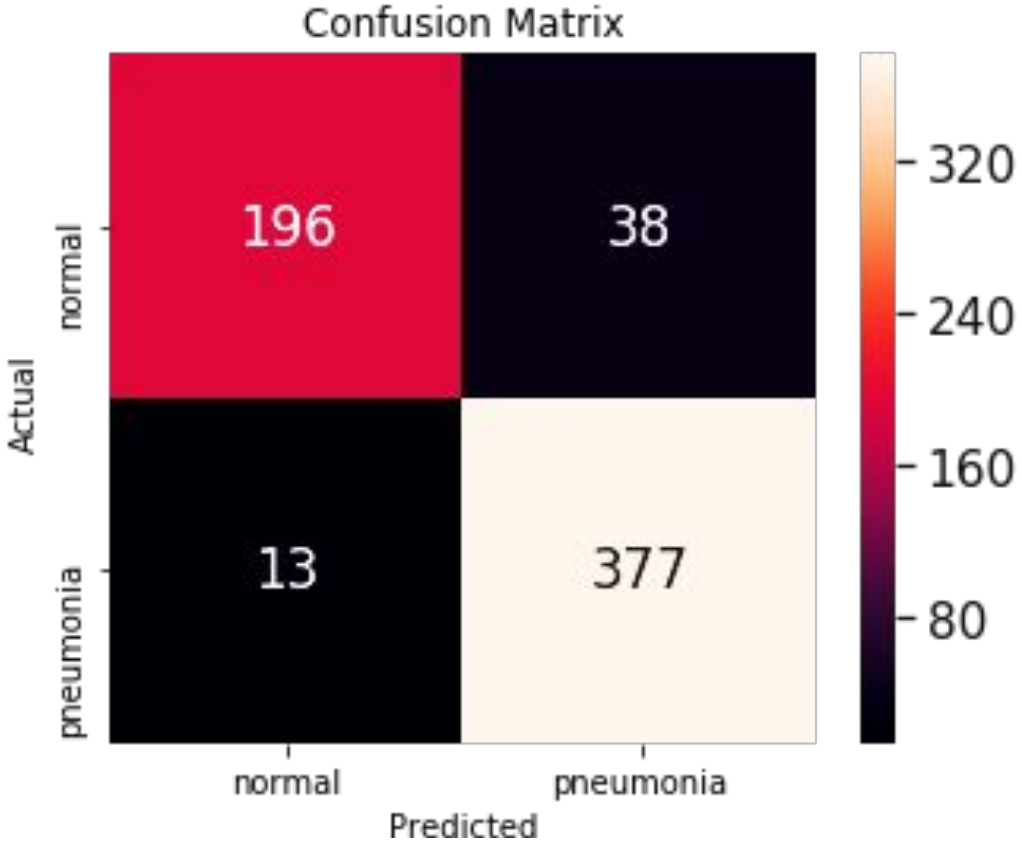

```
Epoch 1/10
163/163 [==============================] - 109s 672ms/step - loss: 0.3907 - accuracy: 0.8250 - val_loss: 0.7992 - v
al_accuracy: 0.6234
Epoch 2/10
163/163 [==============================] - 99s 605ms/step - loss: 0.2798 - accuracy: 0.8871 - val_loss: 1.3151 - va
l_accuracy: 0.6267
Epoch 3/10
163/163 [==============================] - 95s 584ms/step - loss: 0.2403 - accuracy: 0.9049 - val_loss: 2.7842 - va
l_accuracy: 0.6233
Epoch 4/10
163/163 [==============================] - 94s 579ms/step - loss: 0.2201 - accuracy: 0.9172 - val_loss: 2.6191 - va
l_accuracy: 0.6284
Epoch 5/10
163/163 [==============================] - 93s 570ms/step - loss: 0.2144 - accuracy: 0.9212 - val_loss: 0.2947 - va
l_accuracy: 0.8699

Epoch 00005: ReduceLROnPlateau reducing learning rate to 0.0003000000142492354.
Epoch 6/10
163/163 [==============================] - 93s 571ms/step - loss: 0.1771 - accuracy: 0.9321 - val_loss: 0.1667 - va
l_accuracy: 0.8834
Epoch 7/10
163/163 [==============================] - 93s 571ms/step - loss: 0.1546 - accuracy: 0.9477 - val_loss: 0.5263 - va
l_accuracy: 0.8074

Epoch 00007: ReduceLROnPlateau reducing learning rate to 9.000000427477062e-05.
Epoch 8/10
163/163 [==============================] - 93s 571ms/step - loss: 0.1448 - accuracy: 0.9452 - val_loss: 0.2665 - va
l_accuracy: 0.9054
Epoch 9/10
163/163 [==============================] - 92s 566ms/step - loss: 0.1294 - accuracy: 0.9551 - val_loss: 0.7233 - va
l_accuracy: 0.9003

Epoch 00009: ReduceLROnPlateau reducing learning rate to 2.700000040931627e-05.
Epoch 10/10
163/163 [==============================] - 93s 569ms/step - loss: 0.1244 - accuracy: 0.9557 - val_loss: 0.4203 - va
l_accuracy: 0.9206
```

# Method 1

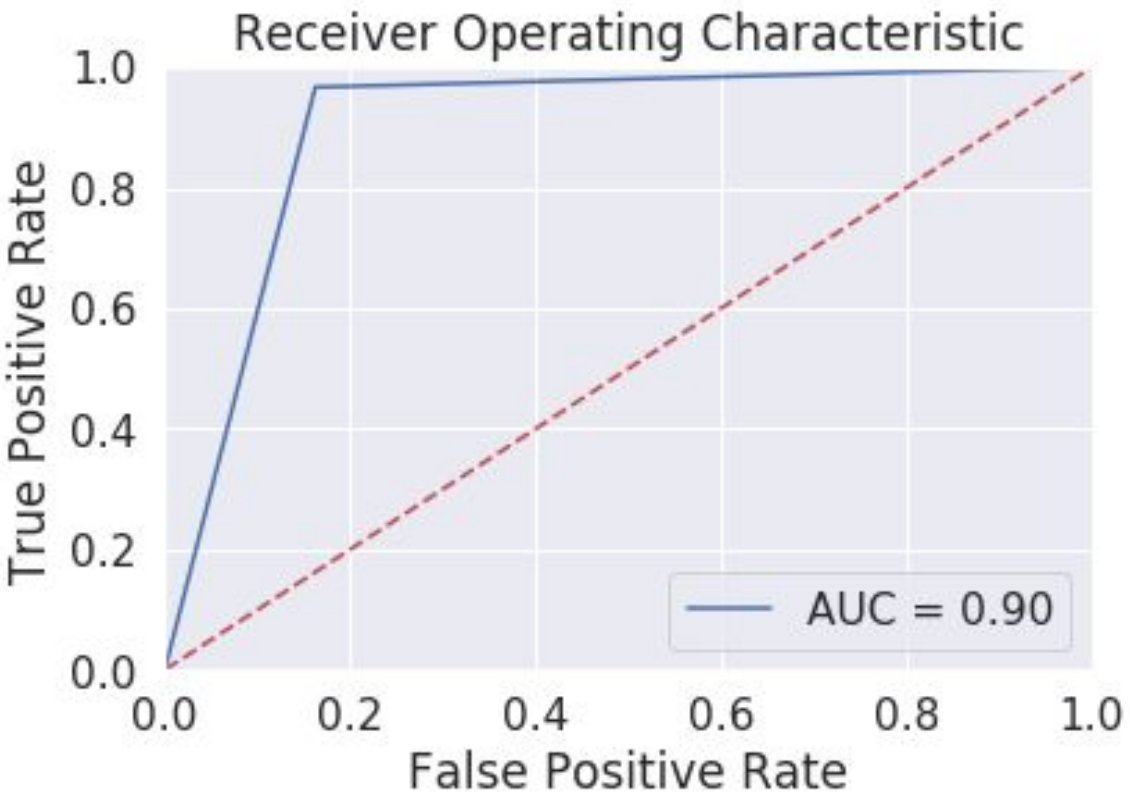

Model accuracy
accuracy
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0
2
4
6
8
epochs
train
val


Model loss
loss
2.5
2.0
1.5
1.0
0.5
0.0
0
2
4
6
8
epochs
train
val

# Method 1


Confusion Matrix
196
38
13
377
normal
pneumonia
Actual
normal
pneumonia
Predicted
320
240
160
80

# Method 1




```
TEST METRICS ----------------------
Accuracy: 91.82692307692307%
Precision: 90.8433734939759%
Recall: 96.66666666666667%
F1-score: 93.66459627329193

TRAIN METRIC ----------------------
Train acc: 95.57
```

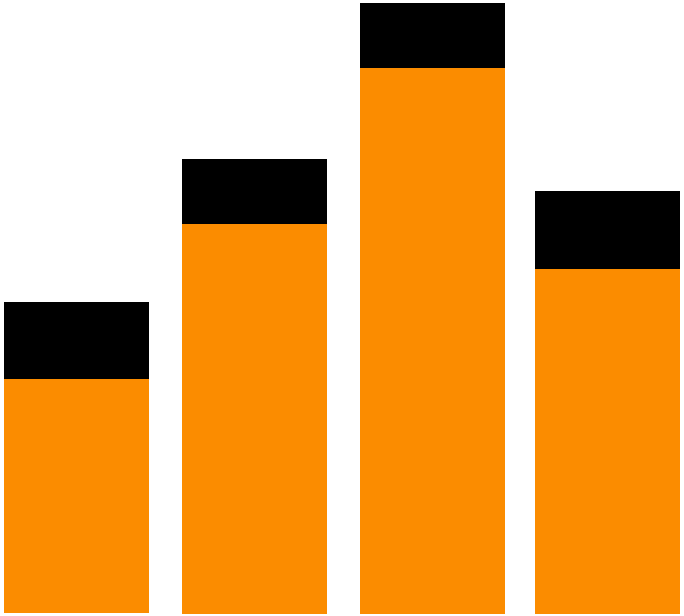
# Analysis

## Method II - Transfer Learning using Keras

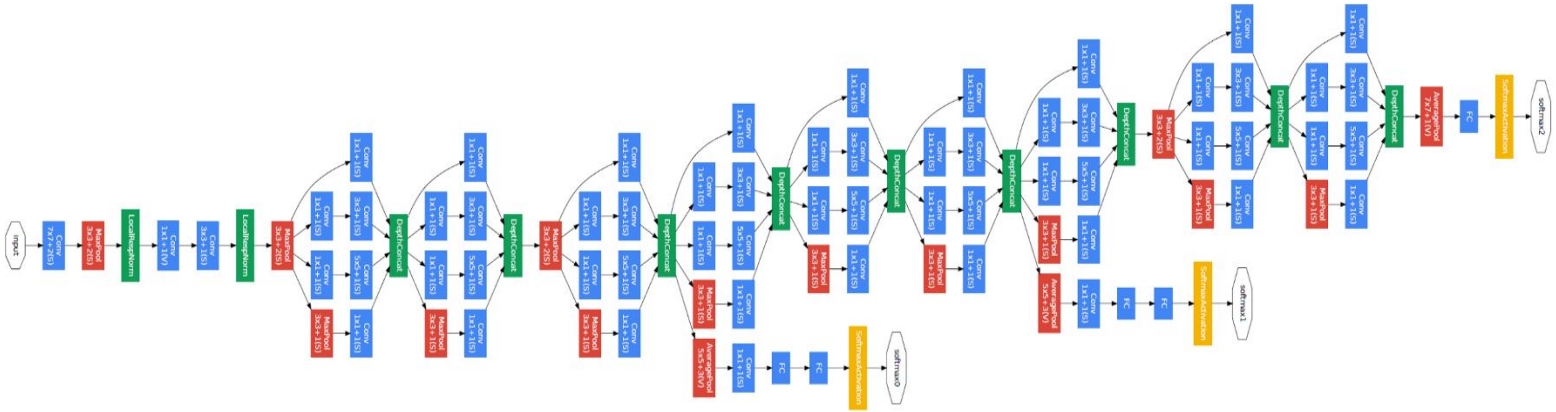
Transfer learning differs from traditional machine learning because it involves using a pretrained model as a catalyst to start a secondary task.

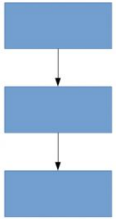
This approach mimics the way humans apply knowledge learned for one task to a new task.

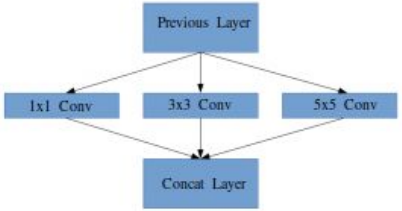
Here we use Inception V3, an award winning model designed by Google used for classification on imageNet dataset.



# Inception V3



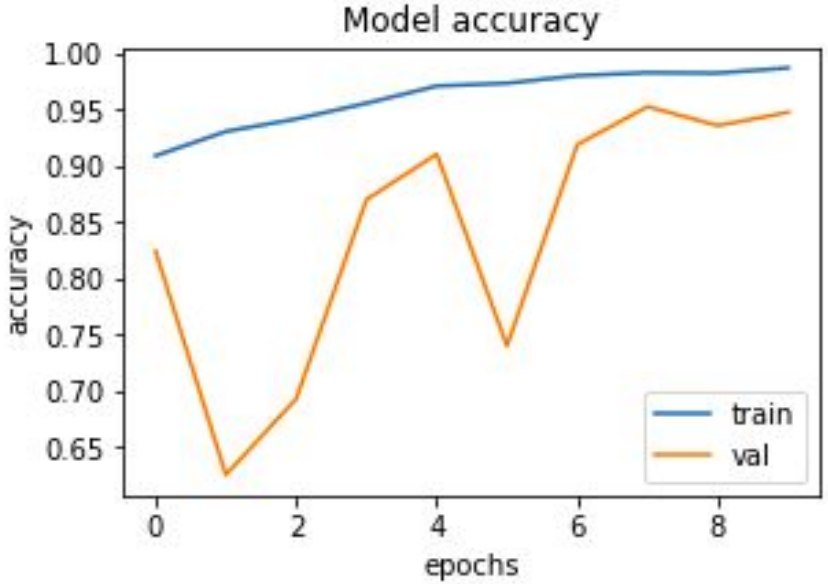


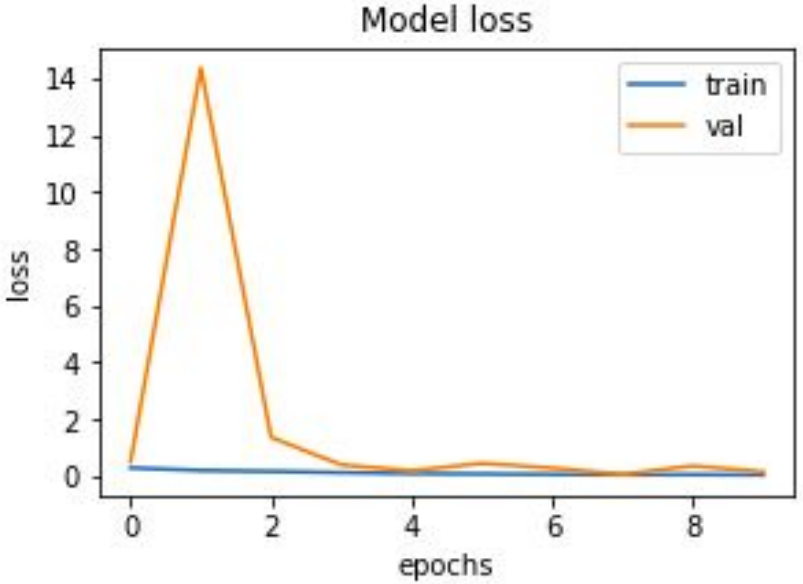



| convolution |
| --- |
| max pool |
| convolution |
| max pool |
| inception (3a) |
| inception (3b) |
| max pool |
| inception (4a) |
| inception (4b) |
| inception (4c) |
| inception (4d) |
| inception (4e) |
| max pool |
| inception (5a) |
| inception (5b) |
| avg pool |
| dropout (40%) |
| linear |
| softmax |

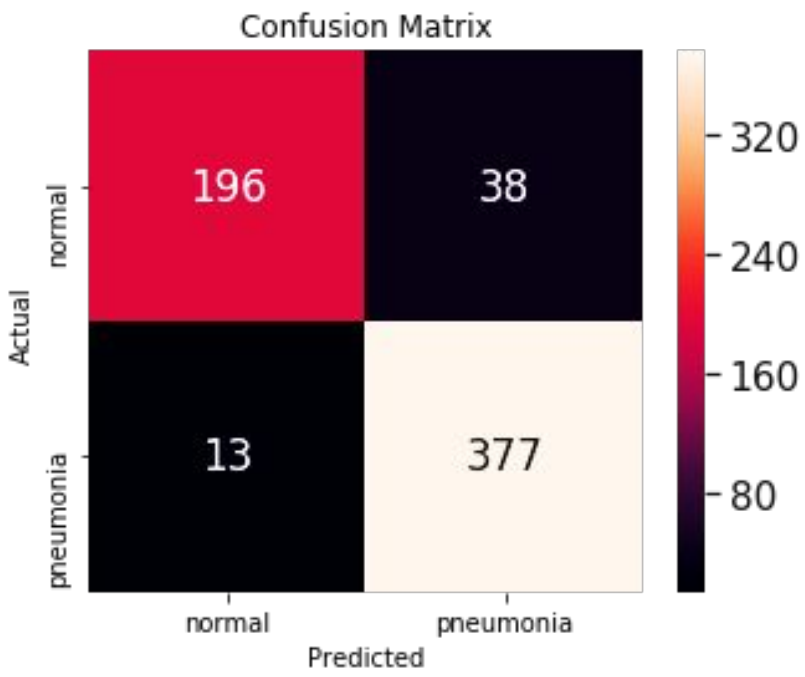
```
==========================================================================================================
Total params: 22,065,697
Trainable params: 22,031,009
Non-trainable params: 34,688
__________________________________________________________________________________________________

Epoch 1/10
163/163 [==============================] - 140s 862ms/step - loss: 0.2863 - accuracy: 0.9089 - val_loss: 0.5051 - v
al_accuracy: 0.8240
Epoch 2/10
163/163 [==============================] - 93s 568ms/step - loss: 0.1943 - accuracy: 0.9306 - val_loss: 14.3677 - v
al_accuracy: 0.6250
Epoch 3/10
163/163 [==============================] - 92s 562ms/step - loss: 0.1624 - accuracy: 0.9417 - val_loss: 1.3599 - va
l_accuracy: 0.6926
Epoch 4/10
163/163 [==============================] - 93s 568ms/step - loss: 0.1202 - accuracy: 0.9557 - val_loss: 0.3897 - va
l_accuracy: 0.8699

Epoch 00004: ReduceLROnPlateau reducing learning rate to 0.0003000000142492354.
Epoch 5/10
163/163 [==============================] - 92s 562ms/step - loss: 0.0838 - accuracy: 0.9711 - val_loss: 0.1818 - va
l_accuracy: 0.9105
Epoch 6/10
163/163 [==============================] - 90s 554ms/step - loss: 0.0733 - accuracy: 0.9735 - val_loss: 0.4462 - va
l_accuracy: 0.7399

Epoch 00006: ReduceLROnPlateau reducing learning rate to 9.000000427477062e-05.
Epoch 7/10
163/163 [==============================] - 93s 568ms/step - loss: 0.0572 - accuracy: 0.9803 - val_loss: 0.2785 - va
l_accuracy: 0.9189
Epoch 8/10
163/163 [==============================] - 92s 564ms/step - loss: 0.0468 - accuracy: 0.9829 - val_loss: 0.0567 - va
l_accuracy: 0.9527

Epoch 00008: ReduceLROnPlateau reducing learning rate to 2.700000040931627e-05.
Epoch 9/10
163/163 [==============================] - 91s 558ms/step - loss: 0.0501 - accuracy: 0.9826 - val_loss: 0.3555 - va
l_accuracy: 0.9358
Epoch 10/10
163/163 [==============================] - 92s 563ms/step - loss: 0.0406 - accuracy: 0.9872 - val_loss: 0.1494 - va
l_accuracy: 0.9476

Epoch 00010: ReduceLROnPlateau reducing learning rate to 8.100000013655517e-06.
```

# Method 2

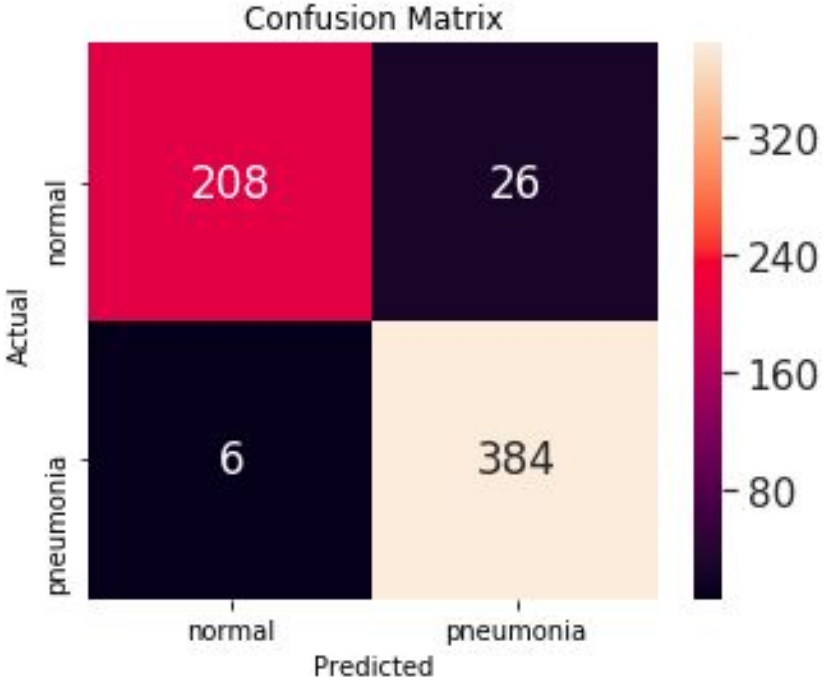

Model accuracy
accuracy
epochs
train
val
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0
2
4
6
8


Model loss
loss
epochs
train
val
14
12
10
8
6
4
2
0
0
2
4
6
8

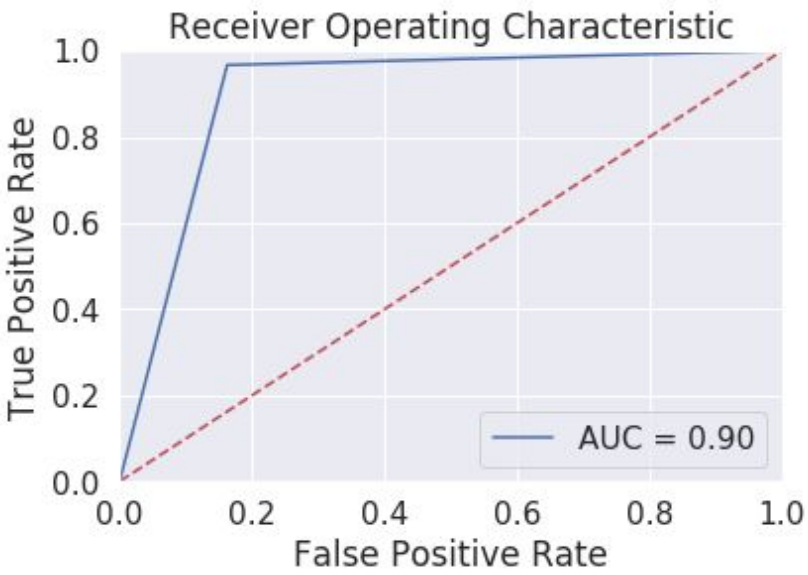
# Method 1




# Method 2

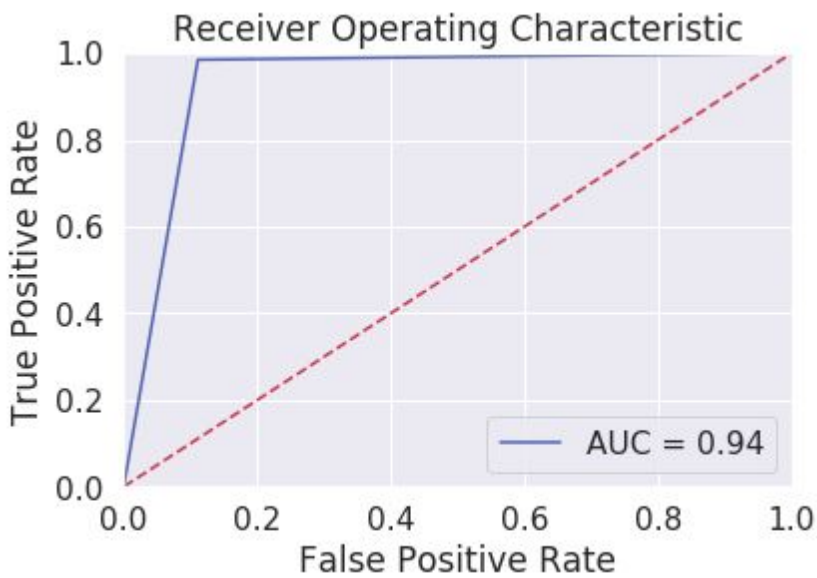


# Method 1




```
TEST METRICS ----------------------
Accuracy: 91.82692307692307%
Precision: 90.8433734939759%
Recall: 96.66666666666667%
F1-score: 93.66459627329193

TRAIN METRIC ----------------------
Train acc: 95.57
```

# Method 2




```
TEST METRICS ----------------------
Accuracy: 94.87179487179486%
Precision: 93.65853658536587%
Recall: 98.46153846153847%
F1-score: 96.00000000000001

TRAIN METRIC ----------------------
Train acc: 98.72
```

# DISCUSSION

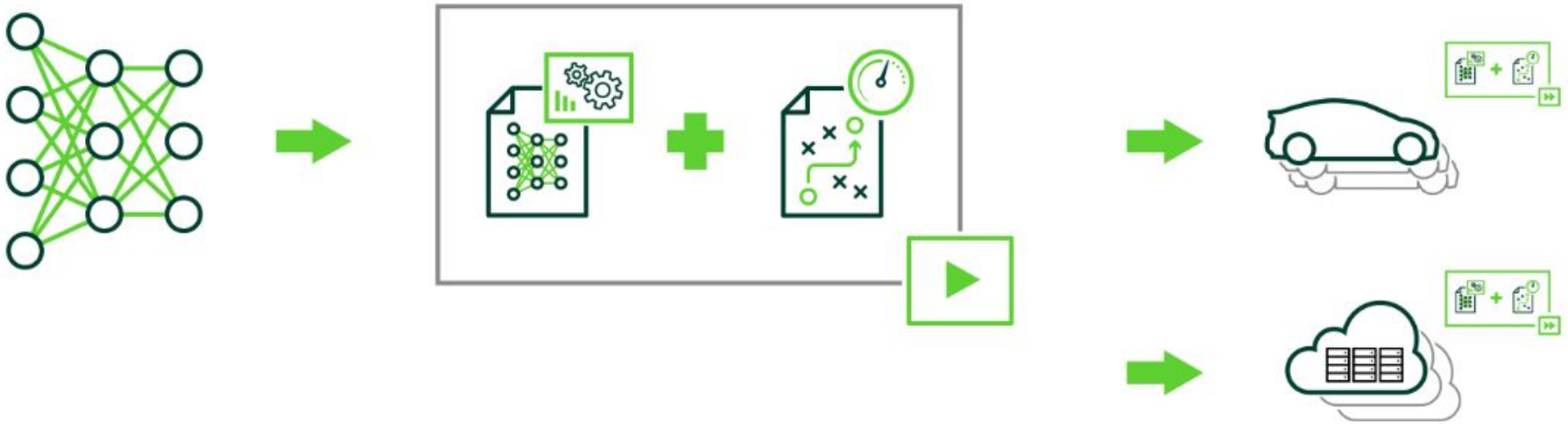
- Computational Challenges [ GPU ]
- Scalability [ Hadoop ] [ Spark ]
- Scope [ Deployment ] [ Generalization ]


# GPU


APACHE
hadoop
TM


APACHE
Spark
TM

# Deployment Generalization



# References


- https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia
- https://www.github.com
- https://towardsdatascience.com
- https://missinglink.ai
- https://stackoverflow.com


print('thank you')