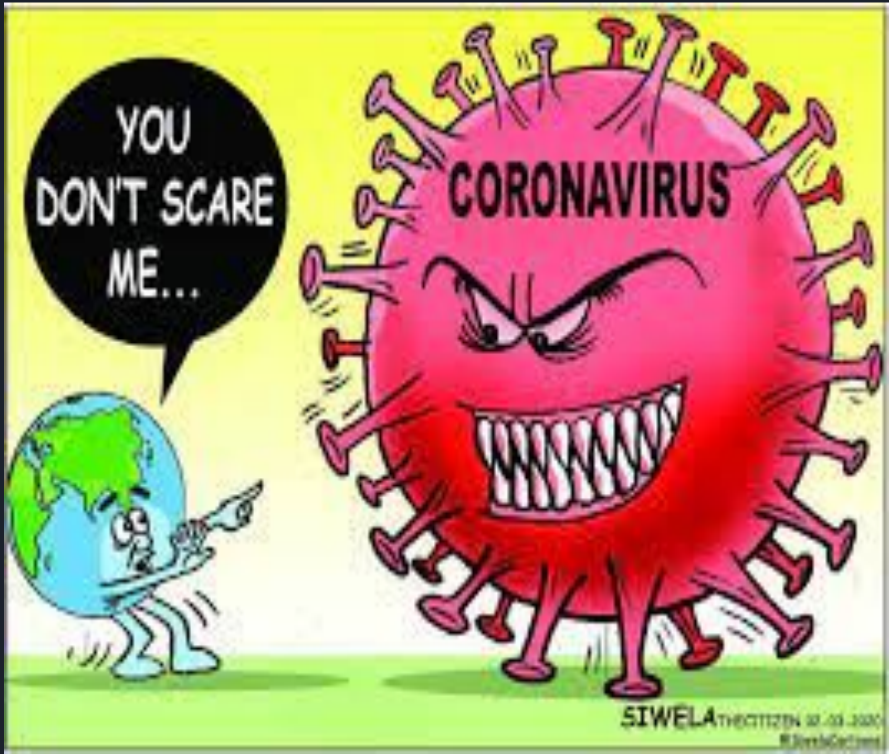
DETECTION OF COVID-19 IN X-RAY IMAGES USING DEEP LEARNING APPROACH (USING CONVOLUTIONAL NEURAL NETWORK)

INDEX

- ABSTRACT
- EXISTING APPROACHES
- PROPOSED MODEL
- INPUT AND OUTPUT EXPECTED
- REQUIREMENTS
- PROJECT FLOW (UPTO REVIEW 1)
- REVIEW 2
- REVIEW 3

ABSTRACT

- COVID19 pandemic has paralyzed the whole world irrespective of any discrimination.
- To reduce the spreading of infection or to eradicate it testing of people plays a key role.
- Deep learning neural networks have an excellent potential for building COVID-19 triage systems and detecting COVID-19 patients, especially patients with low severity.



- ❑ The project aims to understand the accuracy of infection in the people who test for covid 19 with the help of trends and accuracy expected.
- ❑ Furthermore, this project also includes collecting various datasets which consist of X-ray images of many people and training,evaluating models using various deep learning approaches.

EXISTING APPROACHES

- The most related studies to ours, Wang and Wong designed COVID-Net, a deep learning-based model, for COVID19 detection. COVID-Net achieved a 92.4% success rate using a total of 16,756 radiography images obtained from different open access data.

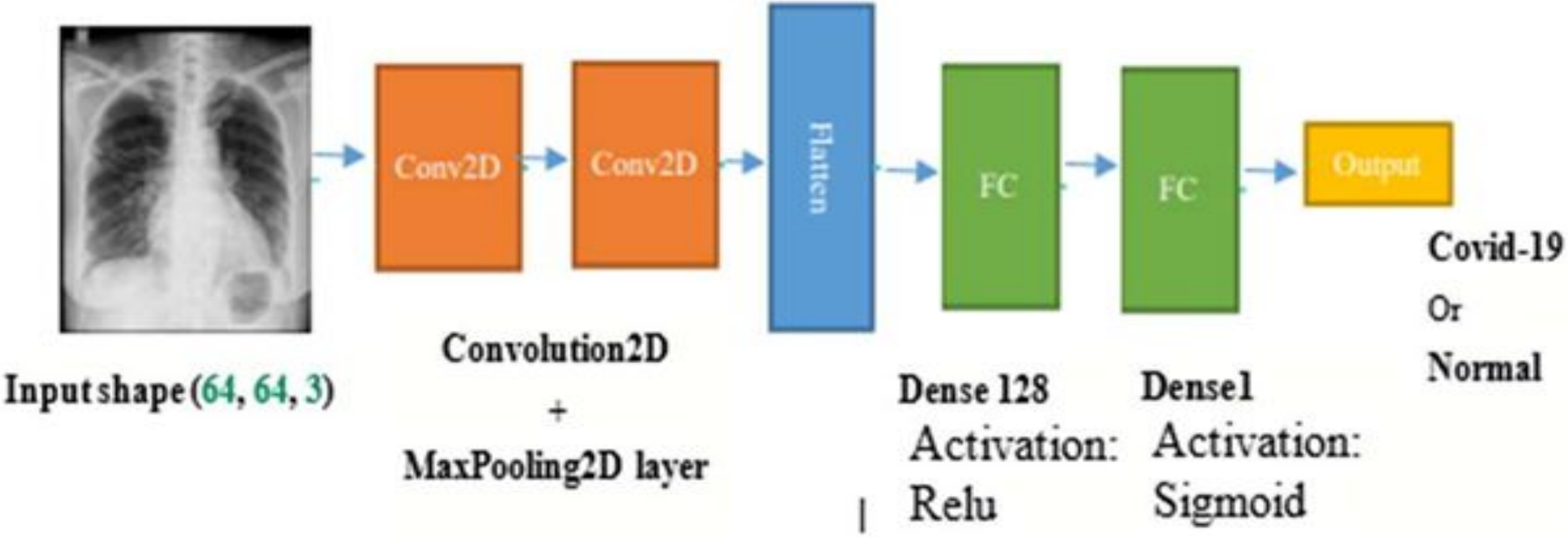
- Hemdan proposed COVIDX-Net to diagnose COVID-19 in X-ray images. They have obtained 90% accuracy rate using 25 COVID-19 positive and 25 normal images.

EXISTING APPROACHES

- Transfer learning, In this study, 224 approved COVID-19, 504 normal radiology images are used. They achieved a 98.75% performance for the 2-class and 93.48% performance for the 3-class problem.

- The authors of a smaller network,called COVID-CAPS, also claim that their model achieved an accuracy of 98.7%, sensitivity of 90%, and specificity of 95.8%.

PROPOSED MODEL



PROPOSED MODEL

CONVOLUTIONAL NEURAL NETWORK :

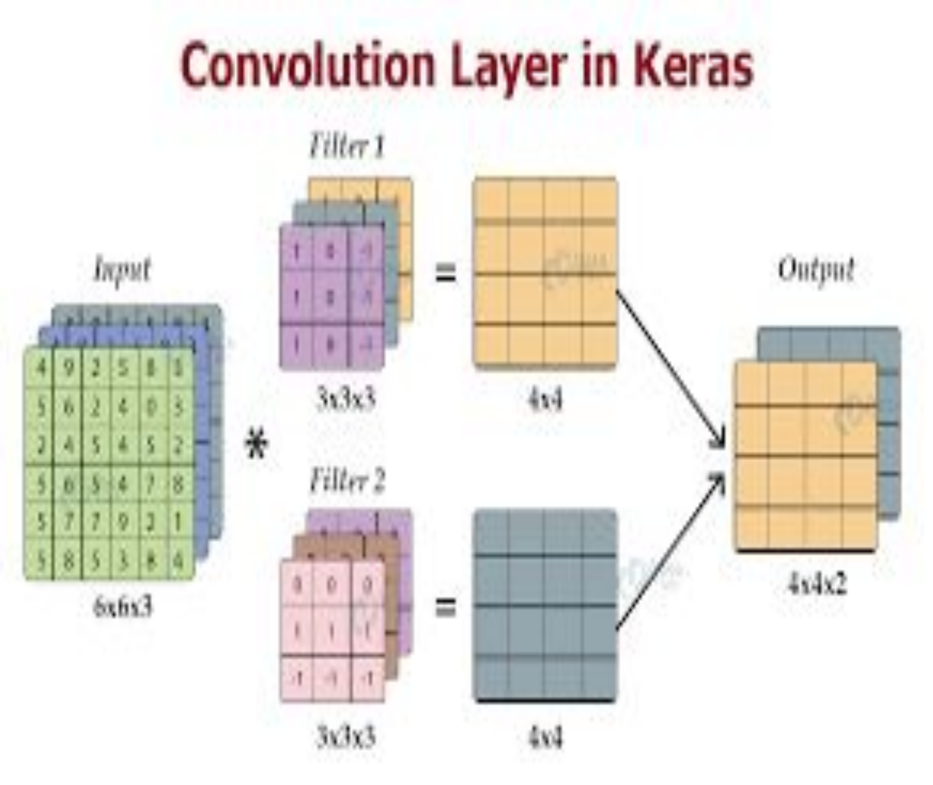
A convolutional neural network (CNN) is a type of artificial neural network used in image recognition and processing that is specifically designed to **process pixel data** .

- It has 3 layers:

  Convolutional layer , pooling layer and fully connected layer

PROPOSED MODEL

- ★ Convolutional layers apply a convolution operation to the input, passing the result to the next layer.
- ★ Max pooling is a pooling operation that selects the maximum element from the region of the feature map covered by the filter. Thus, the output after max-pooling layer would be a feature map containing the most prominent features of the previous feature map.
- ★ Two types: max pooling and average pooling

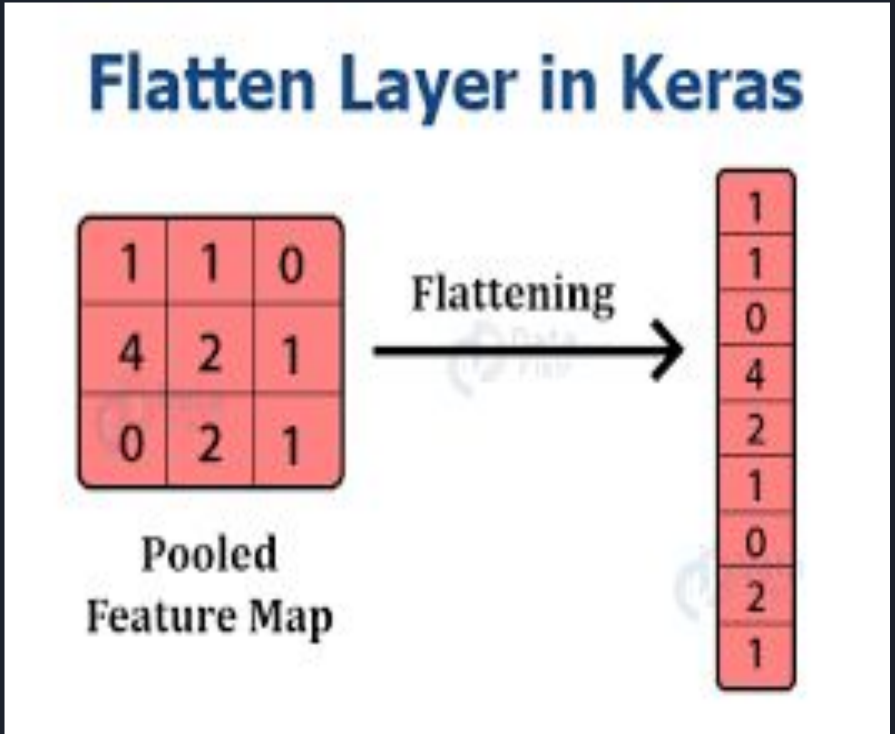


PROPOSED MODEL

- ACTIVATION FUNCTIONS: RELU, SOFTMAX
- Flatten layer:

  Flattening is converting the data into a **1-dimensional array** for inputting it to the next layer.

- Dense layer is **the regular deeply connected neural network layer**.



Flatten Layer in Keras

1	1	0
4	2	1
0	2	1

INPUT AND OUTPUT EXPECTED

INPUT:

- **Images of people with covid-19 positive and negative.**
- **The collected chest X-ray images dataset contains x-ray images divided for training and testing.**
- **DATASETS :**

  http://cb.lk/covid_19

  COVID-19 chest xray | Kaggle

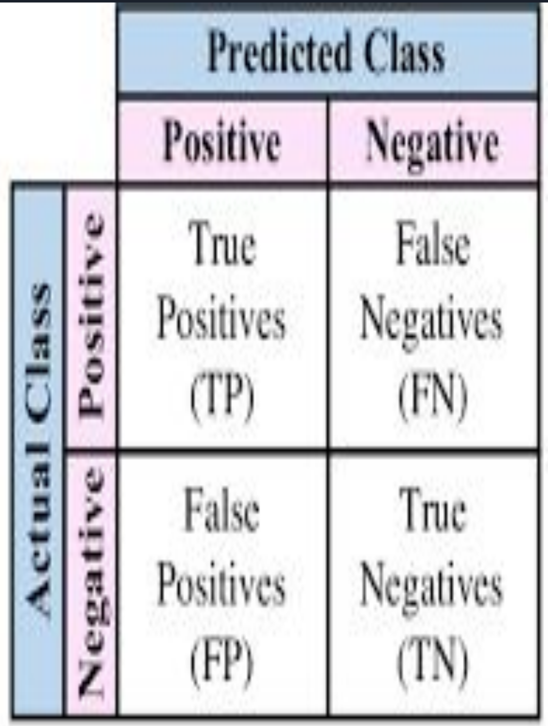
  Chest X-ray (Covid-19 & Pneumonia) | Kaggle

INPUT AND OUTPUT EXPECTED

<u>OUTPUT EXPECTED</u> : CONFUSION MATRIX

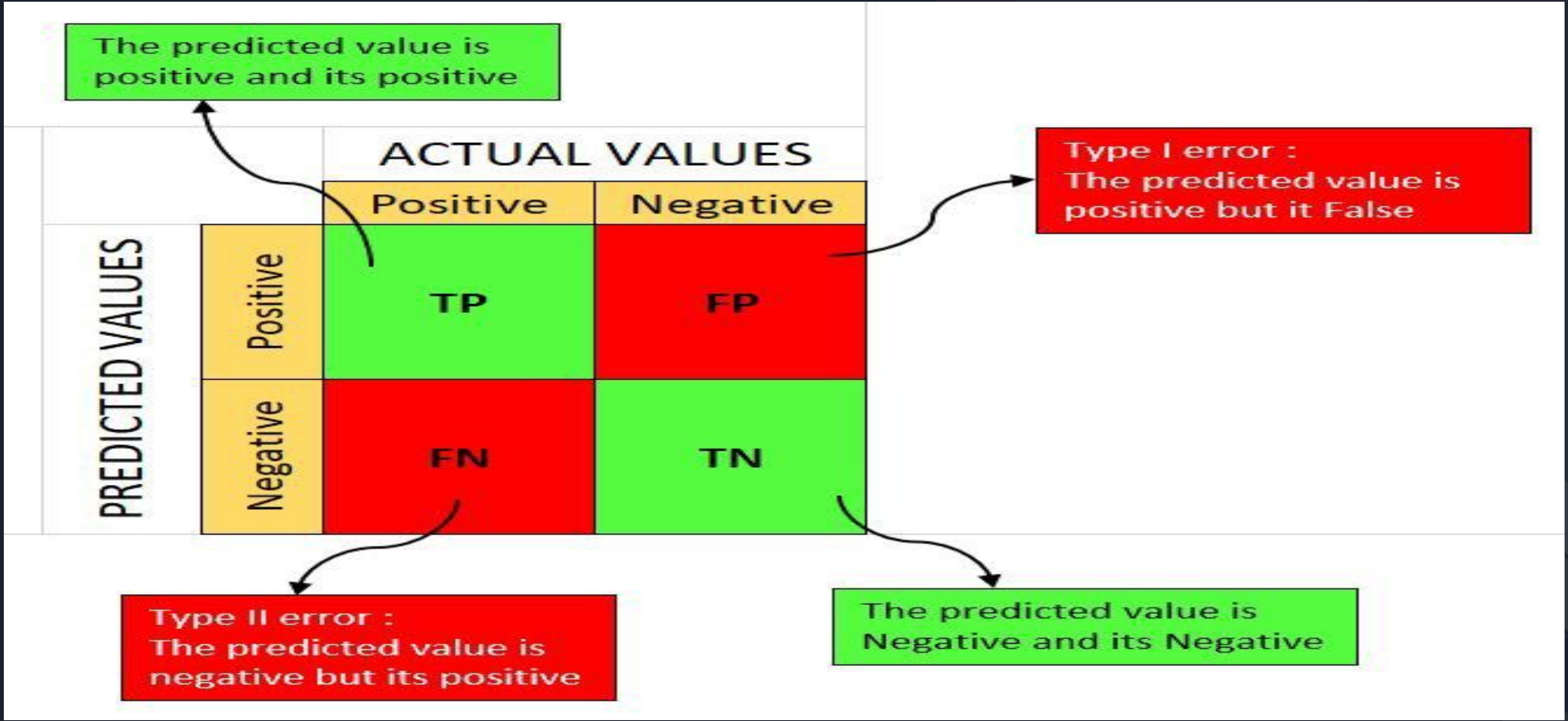
A confusion matrix is a technique for summarizing the performance of classification algorithm.

Classification accuracy alone can be misleading if you have an unequal number of observations in each class or if you have more than two classes in your dataset



		Predicted Class	
		Positive	Negative
Actual Class	Positive	True Positives (TP)	False Negatives (FN)
	Negative	False Positives (FP)	True Negatives (TN)

INPUT AND OUTPUT EXPECTED



The predicted value is positive and its positive

		ACTUAL VALUES	
		Positive	Negative
PREDICTED VALUES	Positive	TP	FP
	Negative	FN	TN

Type I error :
The predicted value is positive but it False

Type II error :
The predicted value is negative but its positive

The predicted value is Negative and its Negative

REQUIREMENTS

- ❑ HARDWARE REQUIREMENTS :
  - Processor: Intel(r) Core(Tm) i5-8300H
  - RAM: 8GB
  - Device: Any device that can access the internet

- ❑ SOFTWARE REQUIREMENTS : GOOGLE COLAB

  Programming languages used: Python3.9

  Python libraries used: keras, Numpy, Matplotlib

Project flow

- The below listed steps explain the flow of project.

1. Collecting datasets
2. Importing necessary libraries

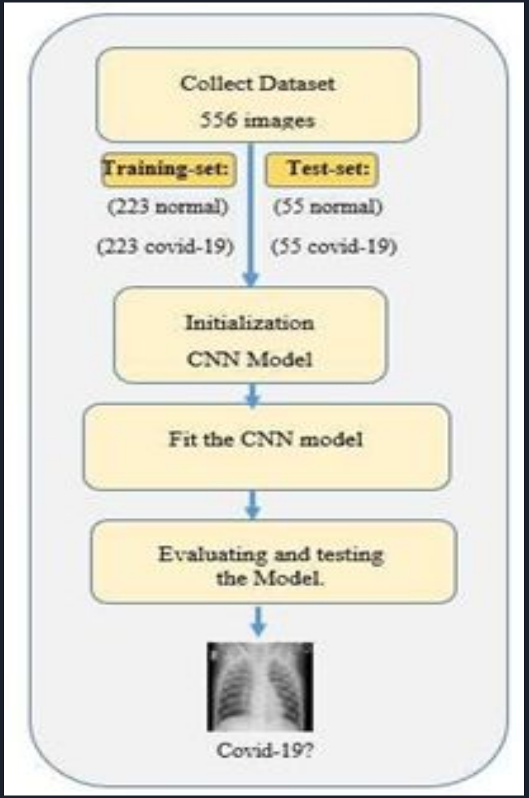
Project Review 1

3. Initialising the CNN model
4. Plotting model summary
5. Data Generation
6. Model Check Point
7. Training the model
8. Loading best weights
9. Saving the model

Project Review 2

10. Plotting accuracy and loss
11. Evaluating the model
12. Testing the model on validation data
13. Final output plotting

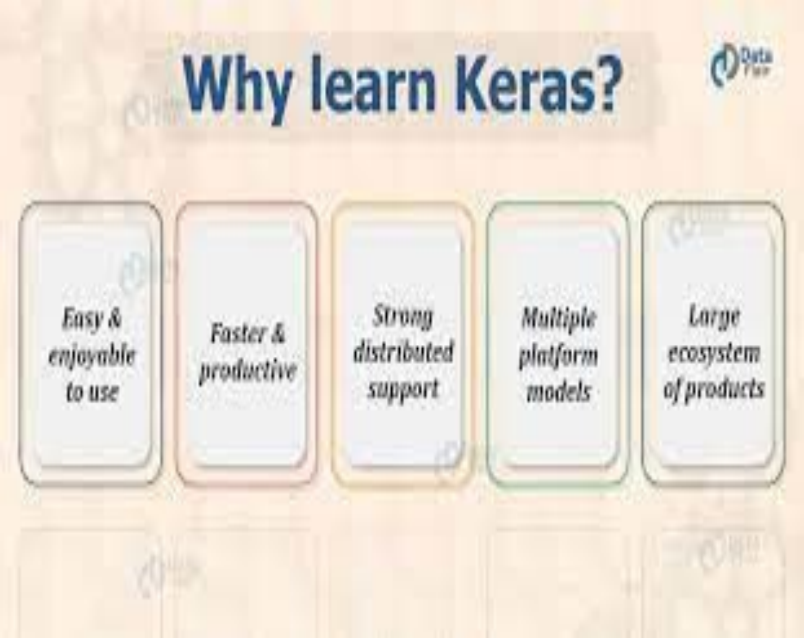
Project review 3



Required Libraries

- **KERAS :** A simplified interface to Tensorflow. Itt is an open-source software library which is built on the top of popular Deep learning libraries like tensorflow etc.. for creating deep learning models is an optimal choice for deep learning applications.

- **MATPLOTLIB & NUMPY :** It is a data visualization and graphical plotting library for python and Numpy is a numerical extension of matplotlib.



INITIALISING CNN MODEL

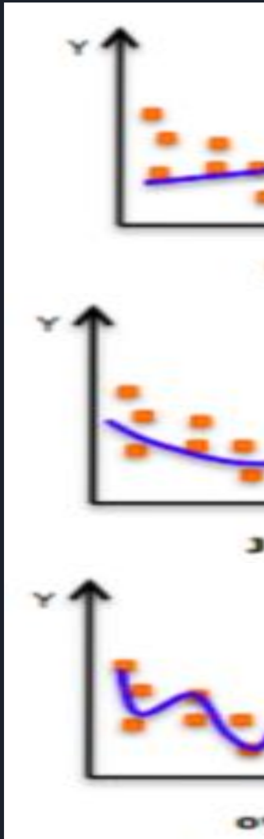
- **conv2D layer :** It is a 2D Convolution Layer, this layer creates a convolution kernel that is wind with layers input which helps produce a tensor of outputs.

- **KERNEL_SIZE :** This parameter determines the dimensions of the kernel. Common dimensions include 1×1, 3×3, 5×5, and 7×7 which can be passed as (1, 1), (3, 3), (5, 5), or (7, 7) tuples.

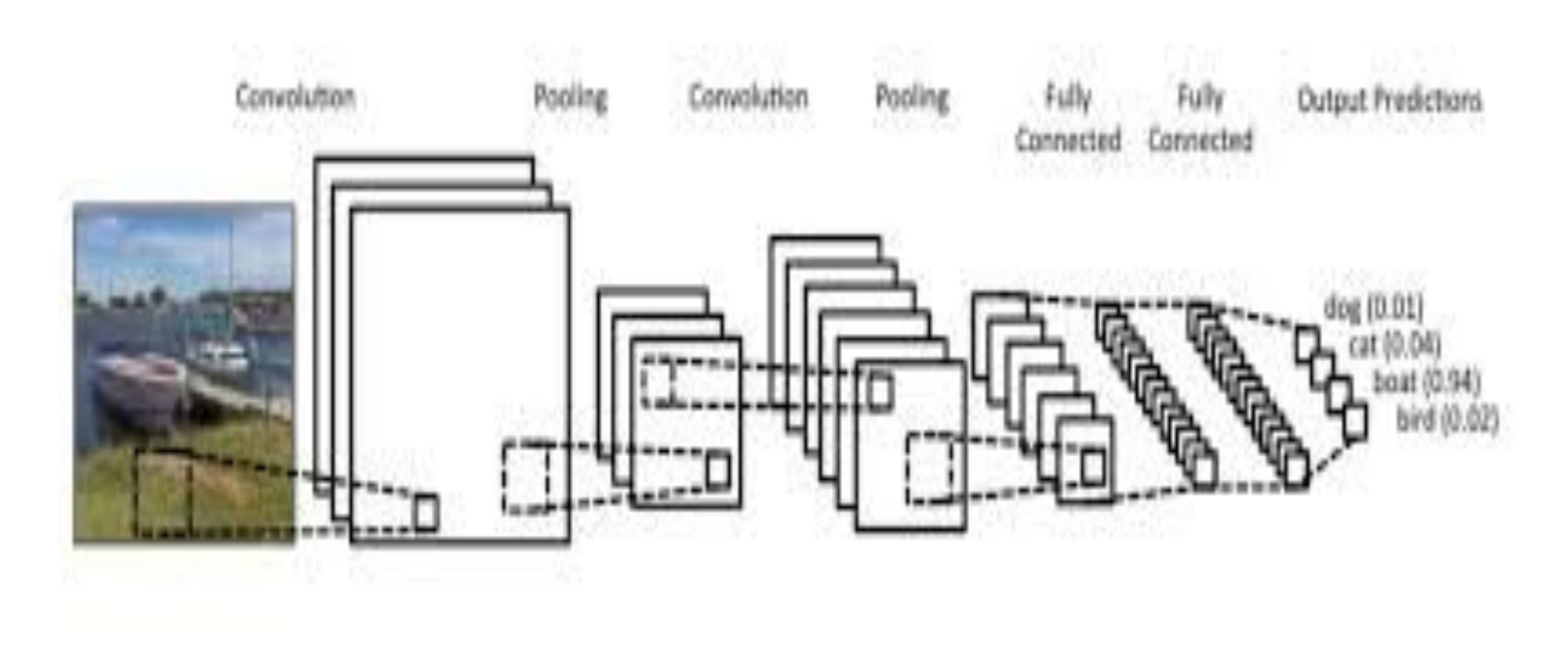
- **ACTIVATION :** The activation parameter to the Conv2D class is simply a convenience parameter which allows you to supply a string, which specifies the name of the activation function you want to apply after performing the convolution. Two types of activation functions are used in the project. They are : RELU ,SIGMOID

- **REGULARIZATION :**
  Regularizations are techniques used to reduce the error by fitting a function appropriately on the given training set and avoid overfitting.

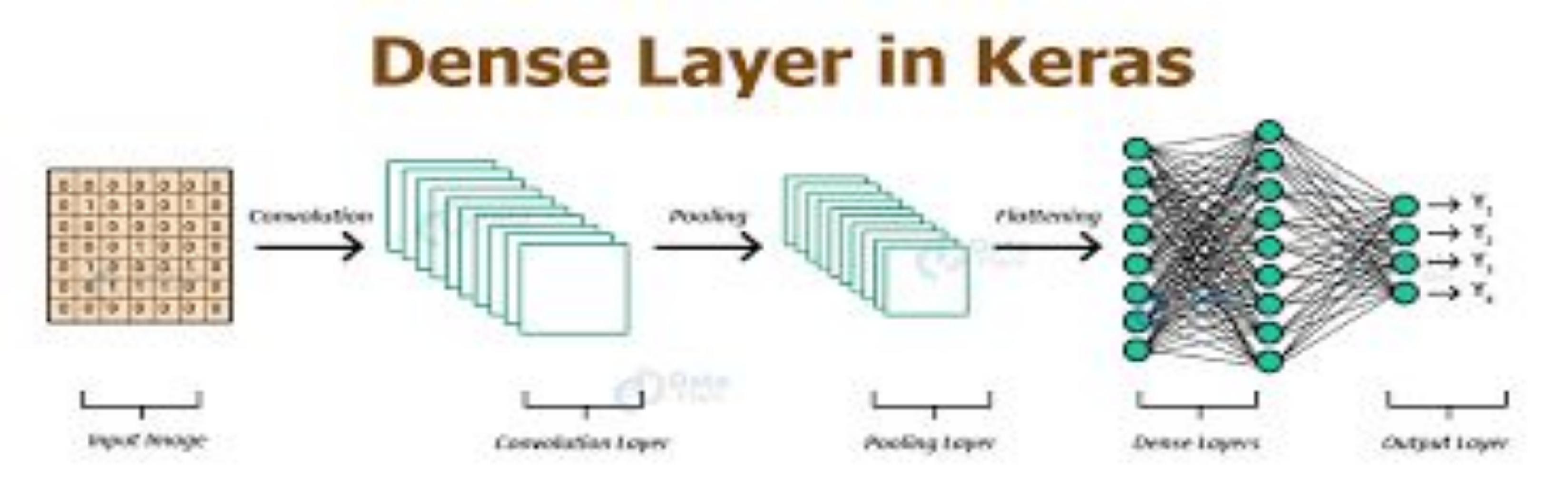
1. Kernel_ regularizer : Regularizer function which is applied to the kernel weights matrix.
2. Bias _ regularizer : Regularizer function which is applied to the bias vector.



- **MAX POOLING :** Selects maximum element from the region of feature map covered by the filter. It extracts the most prominent features if the previous feature map.
- **FLATTEN() :** Used to flatten the dimensions of the image obtained after convolving it.
- **DENSE() :** Used to make this a fully connected model and is the hidden layer.



- **OPTIMIZER :** Important process with optimizer the input weights by comparing prediction and the loss function

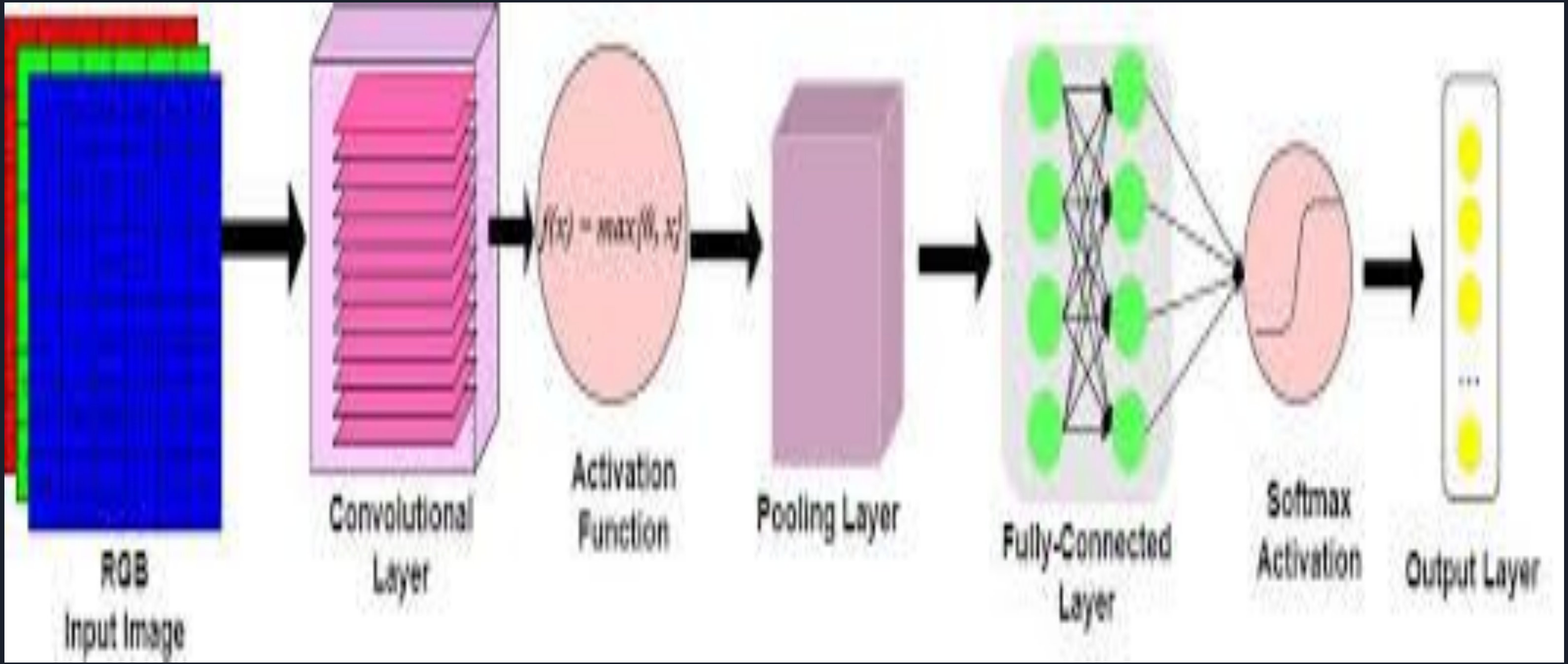


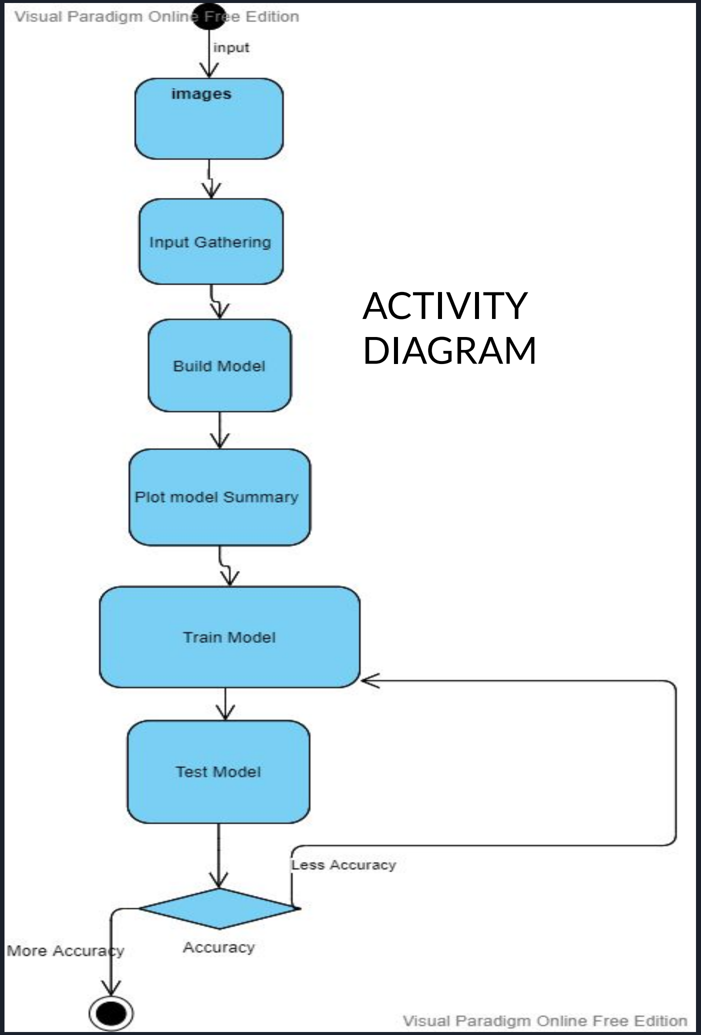
- **ImageDataGenerator :** Keras ImageDataGenerator lets you augment your images in real-time while your model is still training.

- **ModelCheckPoint :** The ModelCheckpoint callback class allows you to define where to checkpoint the model weights, how the file should named and under what circumstances to make a checkpoint of the model.

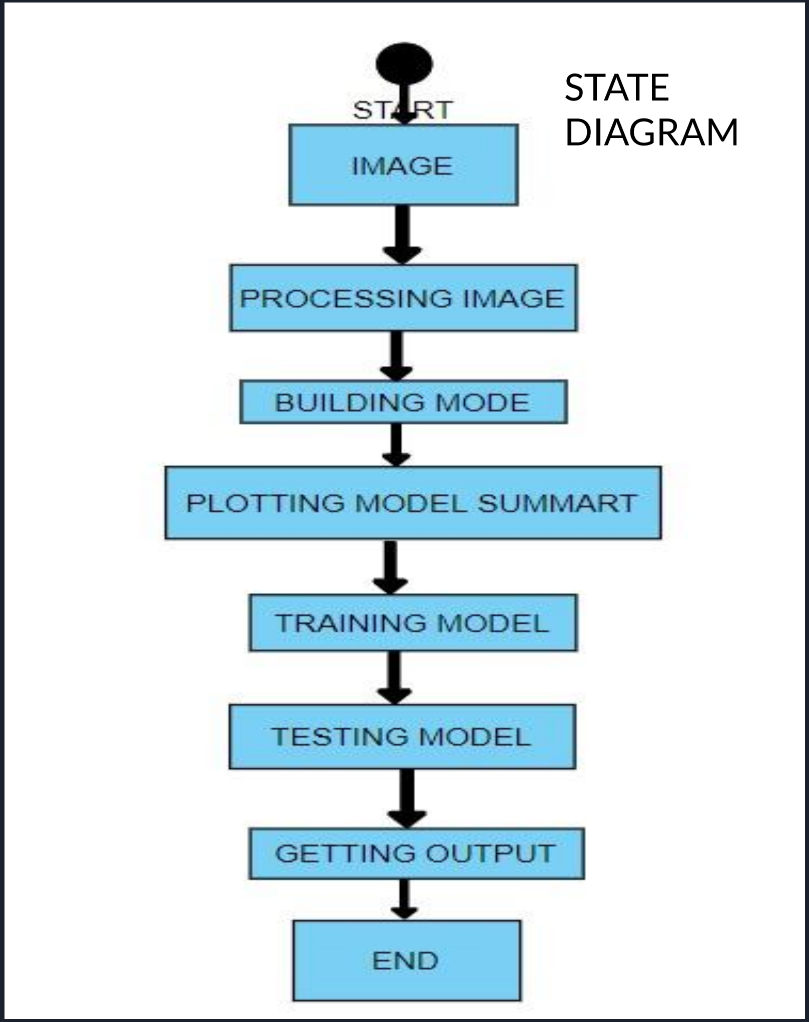
- you can retrieve the values of the weights as a list of Numpy arrays via save_weights(), and set the state of the model via load_weights.

OVERALL VIEW





ACTIVITY DIAGRAM



STATE DIAGRAM

Plotting accuracy and loss:

- Keras provides the capability to register callbacks when training a deep learning model.
- One of the default callbacks that is registered when training all deep learning models is the History callback. It records training metrics for each epoch. This includes the loss and the accuracy (for classification problems) as well as the loss and accuracy for the validation dataset, if one is set.
- The history object is returned from calls to the fit() function used to train the model. Metrics are stored in a dictionary in the history member of the object returned.

Evaluating the model :

- Model Evaluation is the subsidiary part of the model development process. It is the phase that **is decided whether the model performs better**.

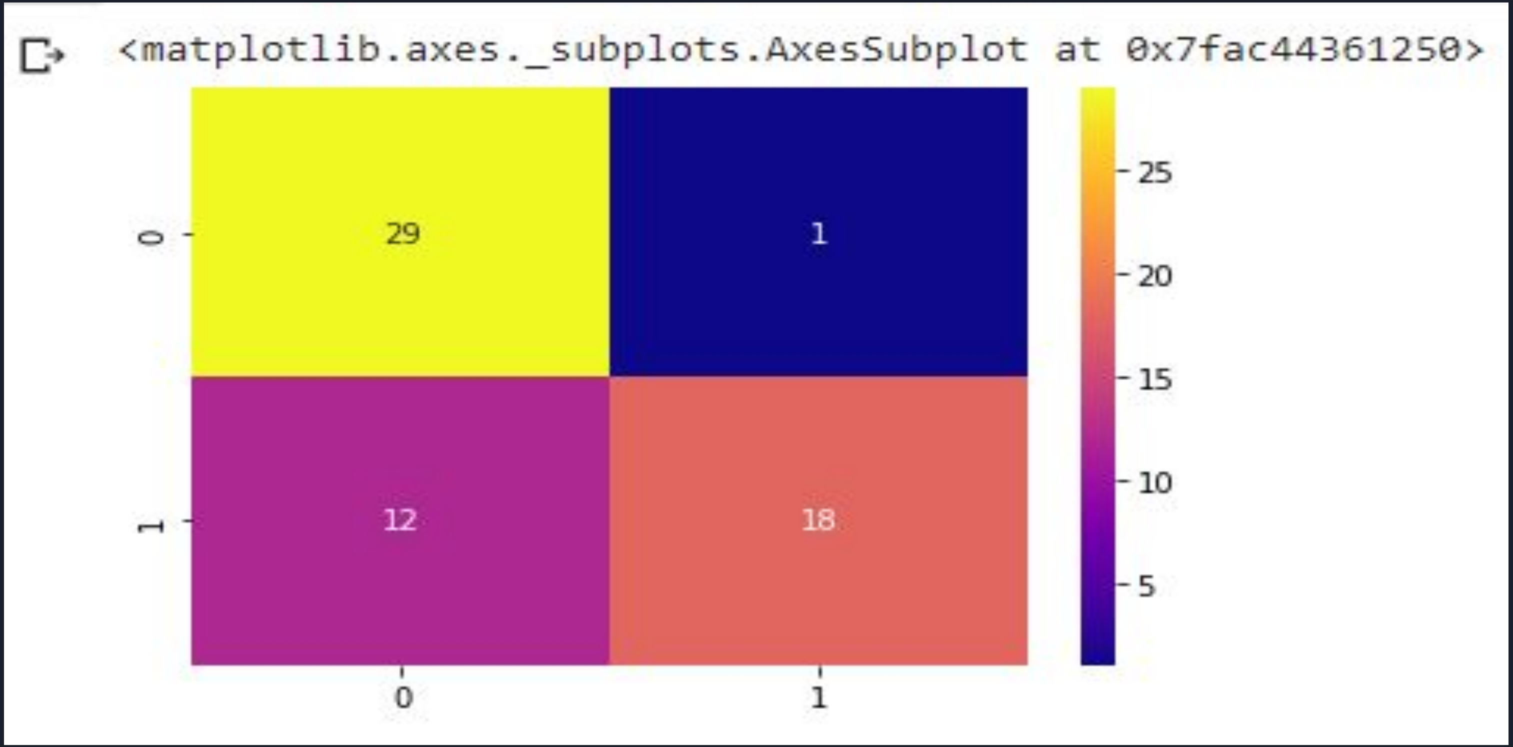
Testing the model on validation data :

- “test” or “validation” data set is **used to qualify performance**.

Final output plotting

- Confusion Matrix is a performance measurement for classification problem where output can be two or more classes.
- A confusion matrix is a summarized table of the number of correct and incorrect predictions (or actual and predicted values) yielded by a classifier (or classification model) for binary classification tasks.

FINAL OUTPUT



```
<matplotlib.axes._subplots.AxesSubplot at 0x7fac44361250>
```

REFERENCES

- P. Rajpurkar et al., “CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning,” pp. 3–9, 2017.

- Worldometer, “COVID-19 Coronavirus Pandemic,” 2020. [Online]. Available: https://www.worldometers.info/coronavirus/?utm_campaign=homeAd vegas1? [Accessed: 28-Nov-2020].

- A. Y. Maghari and J. H. Zendah, “Detecting Significant Events in Arabic Microblogs using Soft Frequent Pattern Mining,” J. Eng. Res. Technol., vol. 6, no. 1, pp. 11–19, 2019.

- X. Mei et al., “Artificial intelligence-enabled rapid diagnosis of patients with COVID-19.,” Nat. Med., 2020, doi: 10.1038/s41591-020-0931-3.

- D. Caruso et al., “Chest CT features of COVID-19 in Rome, Italy,”

Thank You