# COVID-19 DETECTION WITH CHEST X-RAY DATASET USING CNN


**PROJECT BY:**

SAI MOHITH CHINNARI (1604-18-735-086)

DARAPANENI ANUTHAM SIVA KRISHNA(1604-18-735-083)

PAPULWAR OM DATHATREYA (1604-18-735-085)

**UNDER THE GUIDANCE OF:**

MRS.KAMALA KUMARI

ASSISTANT PROFESSOR, ECED

MJCET, HYDERABAD

# OVERVIEW

- INTRODUCTION AND APPLICATION AREAS OF CONVOLUTIONAL NEURAL NETWORKS.
- CURRENT METHODS BEING USED TO DETECT COVID AND HOW OTHER METHODS CAN BE USED TO IMPROVE DIAGNOSIS.
- THE DEFINITION OF CNN, ITS ADVANTAGES AND THE VARIOUS STEPS INVOLVED IN CNN.
- METHODOLOGY
- BLOCK DIAGRAM AND ITS EXPLANATION
- PROJECT TIME LINE DIAGRAM
- REFERENCES

# INTRODUCTION

- THE CORONAVIRUS PANDEMIC THAT HAS SPREAD ACROSS THE WORLD HAS PLACED ALL SECTORS ON LOCKDOWN FOR MANY MONTHS. MANY DIFFERENT APPROACHES HAVE BEEN TAKEN TOWARDS DETECTING COVID-19.
- ON SUCH METHOD IS TO DIAGNOSE COVID-19 INFECTIONS USING RADIOLOGICAL IMAGES SUCH AS X-RAYS.
- THIS PROJECT IS FOCUSED ON MAKING A DEEP LEARNING BASED SYSTEM WHICH UTILIZES CNN TO DETECT COVID-19 FROM XRAY IMAGES
- IT IS CAPABLE OF LEARNING FROM IMPERATIVE EXPERIENCES WITH LONG TERM STATES.

# APPLICATION AREAS

- EARLY AND EFFICIENT DETECTION OF COVID-19
- SMART HEALTH CARE SYSTEMS
- DIAGNOSIS OF SEVERAL AILMENTS IN THE HUMAN BODY
- DETECTION OF TUMOR REGIONS IN THE LUNGS
- DETECTION OF X-RAY BONE SUPPRESSION

# LITERATURE SURVEY:

**COVID-NET: A TAILORED DEEP CONVOLUTIONAL NEURAL NETWORK DESIGN FOR DETECTION OF COVID-19 CASES FROM CHEST X-RAY IMAGES. (WANG L., LIN Z.Q. AND WONG A):**

- MOTIVATED BY THE NEED FOR FASTER INTERPRETATION OF RADIOGRAPHY IMAGES, A NUMBER OF ARTIFICIAL INTELLIGENCE (AI) SYSTEMS BASED ON DEEP LEARNING21 HAVE BEEN PROPOSED AND RESULTS HAVE SHOWN TO BE QUITE PROMISING IN TERMS OF ACCURACY IN DETECTING PATIENTS INFECTED WITH COVID-19 VIA RADIOGRAPHY IMAGING

**DETECTION OF CORONAVIRUS DISEASE (COVID-19) BASED ON DEEP FEATURES. (KUMAR P. AND KUMARI S).**

- THE SUPPORT VECTOR MACHINE CLASSIFIES THE CORONA AFFECTED X-RAY IMAGES FROM OTHERS USING THE DEEP FEATURE. THE METHODOLOGY IS BENEFICIAL FOR THE MEDICAL PRACTITIONER FOR DIAGNOSIS OF CORONAVIRUS INFECTED PATIENT

**AUTOMATED DETECTION OF COVID-19 CASES USING DEEP NEURAL NETWORKS WITH X-RAY IMAGES. (OZTURK T, TALO M, YILDIRIM EA, BALOGLU UB, YILDIRIM O, ACHARYA UR):**

- RECENT FINDINGS OBTAINED USING RADIOLOGY IMAGING TECHNIQUES SUGGEST THAT SUCH IMAGES CONTAIN SALIENT INFORMATION ABOUT THE COVID-19 VIRUS. APPLICATION OF ADVANCED ARTIFICIAL INTELLIGENCE (AI) TECHNIQUES COUPLED WITH RADIOLOGICAL IMAGING CAN BE HELPFUL FOR THE ACCURATE DETECTION OF THIS DISEASE, AND CAN ALSO BE ASSISTIVE TO OVERCOME THE PROBLEM OF A LACK OF SPECIALIZED PHYSICIANS IN REMOTE VILLAGES.

# DESIGN OBJECTIVE

- CURENTLY COVID-19 IS BEING DETECTED USING **RT-PCR** TESTS WHICH TAKE 4 TO 6 HRS TO GET RESULTS WHICH TAKE LONG TIME COMPARED TO THE COVID-19 RAPID SPREAD RATE IN ADDITION TO BEING INEFFICIENT
- AS A RESULT MANY INFECTED PATIENTS CANNOT BE DETECTED IN TIME AND CAN UNKNOWINGLY INFECT OTHERS
- WITH THE DETECTION OF THIS DISEASE AT AN EARLY STAGE, THE PREVALENCE OF COVID-19 DISEASE WILL DECREASE
- DETECTION OF COVID-19 USING X-RAY IMAGES IS AN ACCURATE AND EFFICIENT METHOD FOR FAST DIAGNOSIS. THIS IS DONE BY USING A DEEP LEARNING BASED SYSTEM WHICH UTILIZES **CNN** TO DETECT COVID-19 FROM XRAY IMAGES

# CONVOLUTIONAL NEURAL NETWORK(CNN)

- CONVOLUTIONAL NEURAL NETWORKS ARE DEEP ARTIFICIAL NEURAL NETWORKS THAT ARE USED PRIMARILY TO CLASSIFY IMAGES, CLUSTER THEM BY SIMILARITY (PHOTO SEARCH),AND PERFORM OBJECT RECOGNITION WITHIN SCENES. BY USE OF **CNN'S** IT CAN IDENTIFY FACES, INDIVIDUALS, ANY SIGNS, TUMORS AND MANY OTHER ASPECTS OF VISUAL DATA.
- THE MAIN ADVANTAGE OF CNN COMPARED TO ITS PREDECESSORS IS THAT IT AUTOMATICALLY DETECTS THE IMPORTANT FEATURES WITHOUT ANY HUMAN SUPERVISION.

## STEPS IN CNN:

**STEP-1:**CONVOLUTION

**STEP-2:**POOLING

**STEP-3:**FLATTENING

**STEP-4:**FULL CONNECTION

# METHODOLOGY

- **SIMULATION USED FOR CODE EXECUTION** :

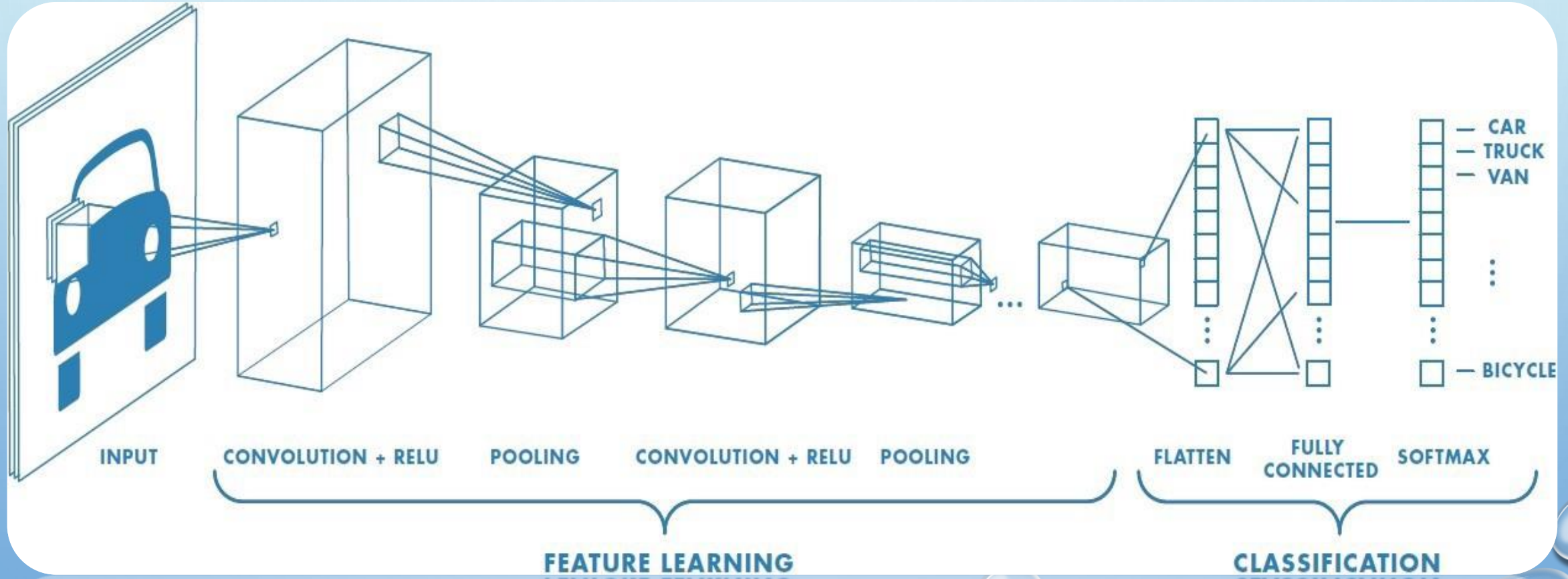
**1.JUPYTER NOTEBOOK:**THE JUPYTER NOTEBOOK IS AN OPEN-SOURCE WEB APPLICATION THAT ALLOWS YOU TO CREATE AND SHARE DOCUMENTS THAT CONTAIN LIVE CODE, EQUATIONS, VISUALIZATIONS AND NARRATIVE TEXT. USES INCLUDE: DATA CLEANING AND TRANSFORMATION, NUMERICAL SIMULATION, STATISTICAL MODELING, DATA VISUALIZATION, MACHINE LEARNING, AND MUCH MORE.

- **LIBRARIES USED:**

**1.NUMPY:** IS A PYTHON LIBRARY USED FOR WORKING WITH ARRAYS. IT ALSO HAS FUNCTIONS FOR WORKING IN DOMAIN OF LINEAR ALGEBRA, FOURIER TRANSFORM,AND MATRICES. IT IS AN EXTENSION OF NUMERIC AND NUMARRAY

**2.TENSORFLOW**: IS A PYTHON LIBRARY FOR FAST NUMERICAL COMPUTING CREATED AND RELEASED BY GOOGLE. IT IS A FOUNDATION LIBRARY THAT CAN BE USED TO CREATE DEEP LEARNING MODELS DIRECTLY OR BY USING WRAPPER LIBRARIES THAT SIMPLIFY THE PROCESS BUILT ON TOP OF TENSORFLOW

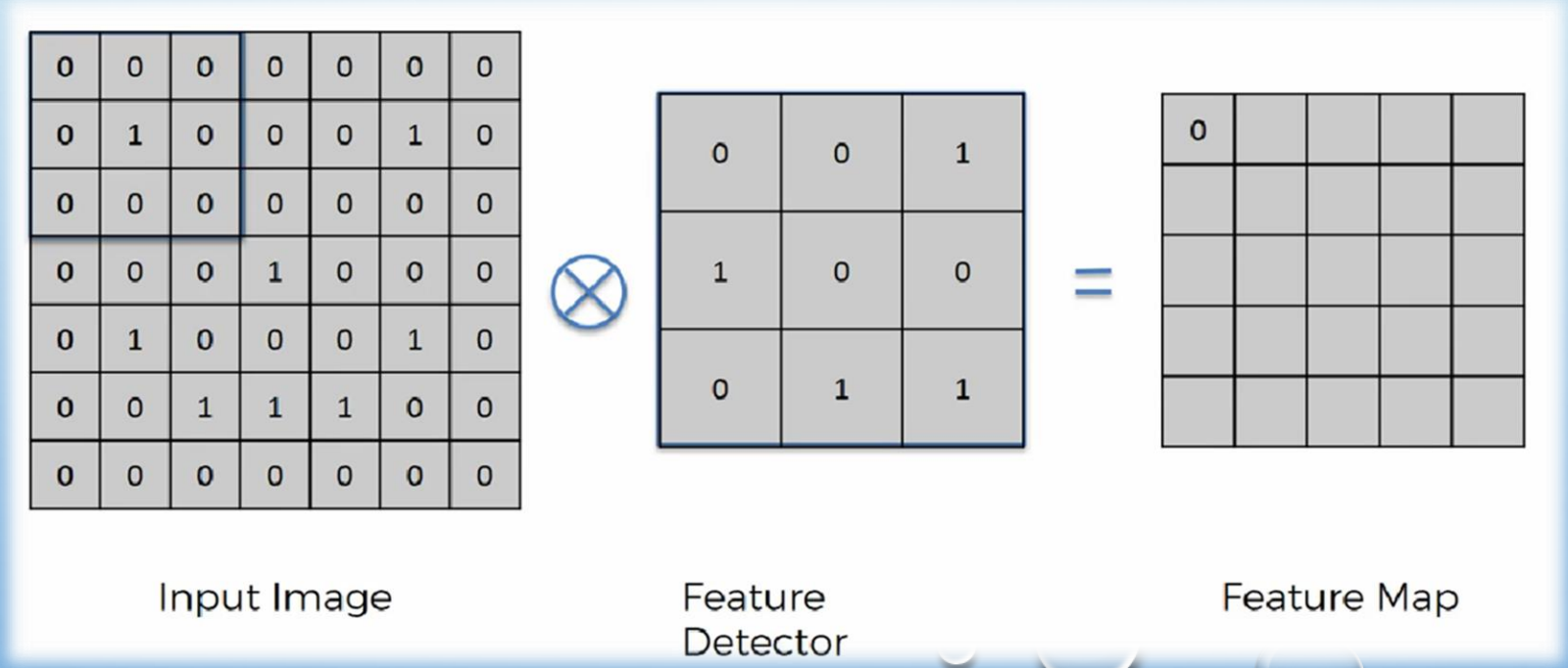
# BLOCK DIAGRAM OF CNN:



# 1(A).CONVOLUTION

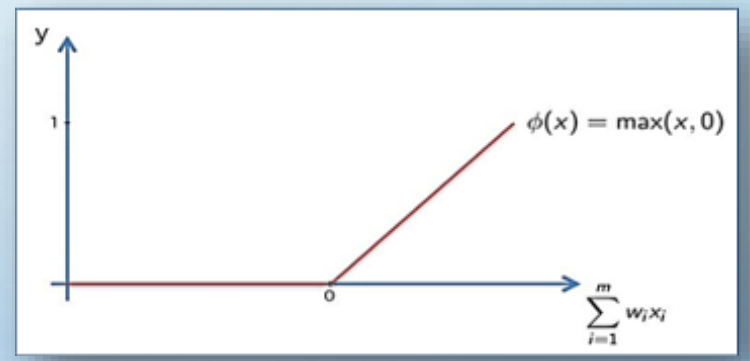
The convolution is performed on the input data with the use of a **filter** or **kernel** (these terms are used interchangeably) to then produce a **feature map.**

We execute a convolution by sliding the filter over the input. At every location, a matrix multiplication is performed and sums the result onto the feature map.



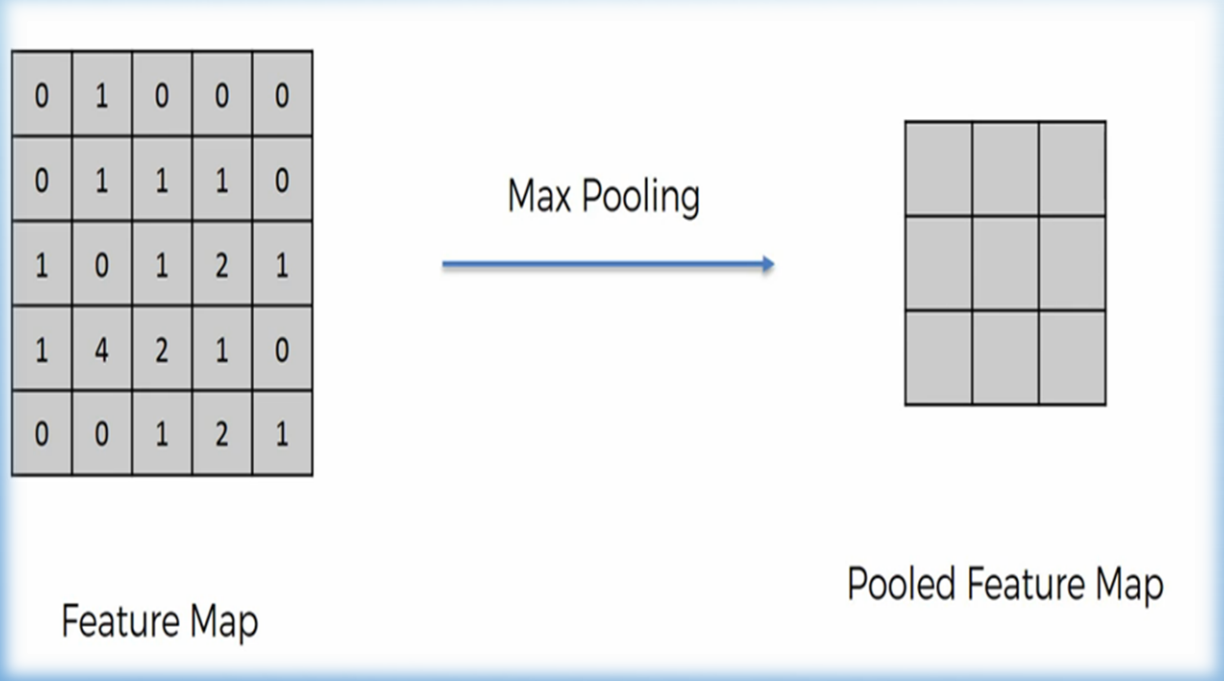
# 1(B). RECTIFIED LINEAR UNIT(RELU)

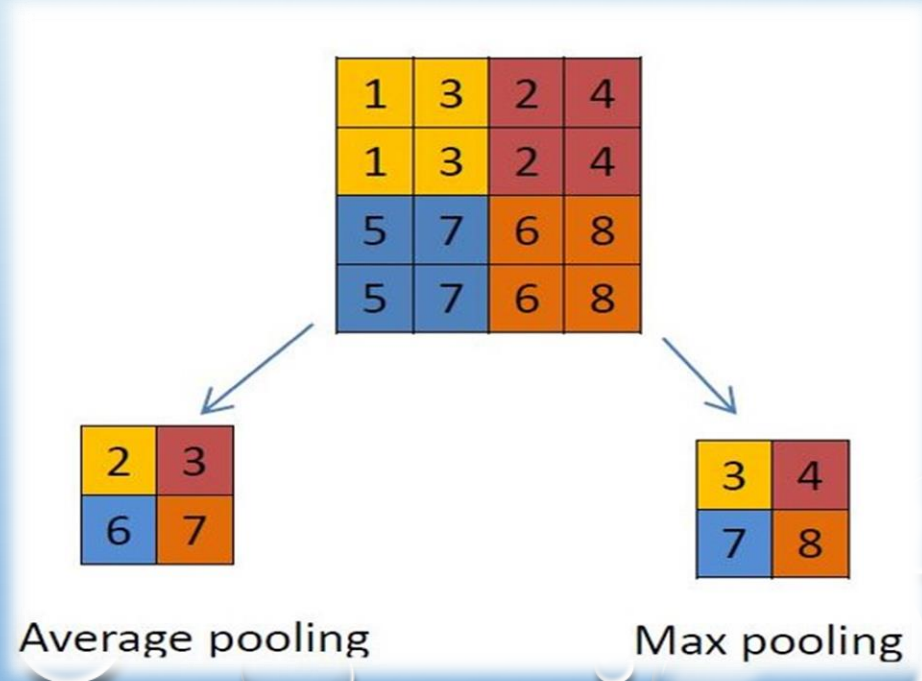
Appling ReLu Activation function to decrease the linearity in the image, because the image is originally non linear



# 2.POOLING

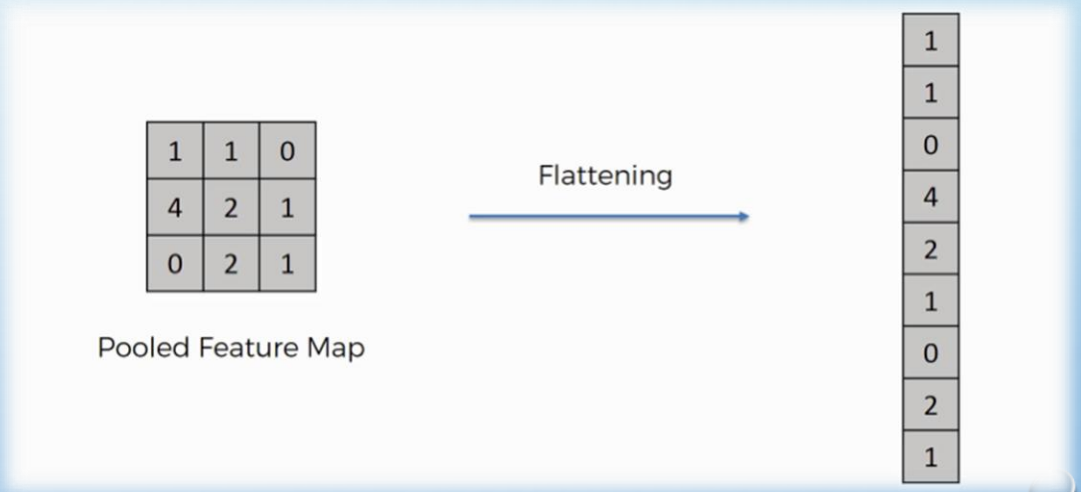
A pooling layer is another building block of a CNN. Its function is to progressively reduce the spatial size of the representation to reduce the amount of parameters and computation in the network. Pooling layer operates on each feature map independently. The most common approach used in pooling is max pooling.



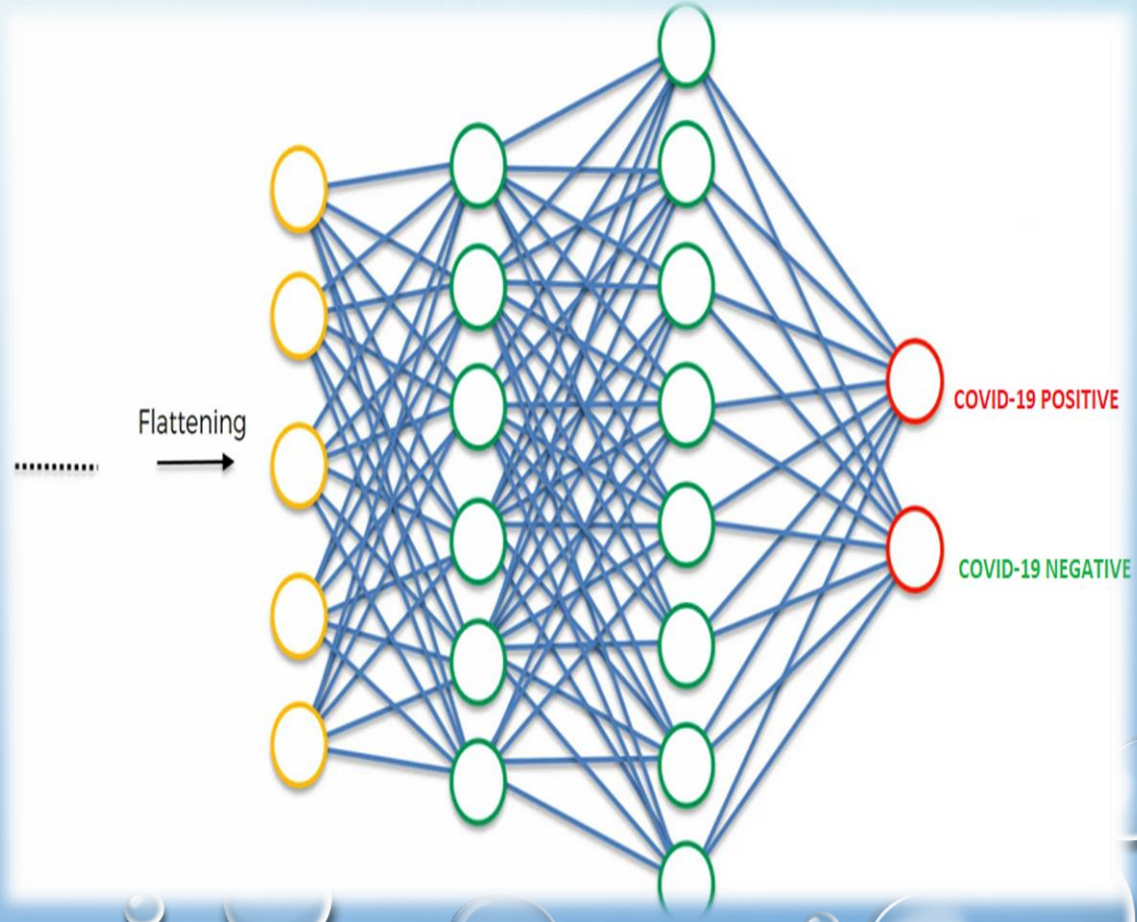
# 3.FLATTENING

we're supposed to have a pooled feature map by now. As the name of this step implies, we are literally going to flatten our pooled feature map into a column like in the image below.The reason we do this is that we're going to need to insert this data into an artificial neural network



# 4.FULL CONNECTION:

THE INPUT LAYER CONTAINS THE VECTOR OF DATA THAT WAS CREATED IN THE FLATTENING STEP. THE FEATURES THAT WE DISTILLED THROUGHOUT THE PREVIOUS STEPS ARE ENCODED IN THIS VECTOR. AT THIS POINT, THEY ARE ALREADY SUFFICIENT FOR A FAIR DEGREE OF ACCURACY IN RECOGNIZING CLASSES. WE NOW WANT TO TAKE IT TO THE NEXT LEVEL IN TERMS OF COMPLEXITY AND PRECISION.



# RESULTS:

- **COVID POSITIVE:**

jupyter Day 5 - Covid Classification - CNN Last Checkpoint: Last Saturday at 19:35 (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

```
7/7 [==============================] - 16s 2s/step - loss: 0.2119 - accuracy: 0.9110 - val_loss: 0.2436 - val_accuracy: 0.8750
Epoch 16/20
7/7 [==============================] - 17s 2s/step - loss: 0.2419 - accuracy: 0.8851 - val_loss: 0.1141 - val_accuracy: 0.9688
Epoch 17/20
7/7 [==============================] - 18s 3s/step - loss: 0.1875 - accuracy: 0.9294 - val_loss: 0.0757 - val_accuracy: 1.0000
Epoch 18/20
7/7 [==============================] - 17s 2s/step - loss: 0.1609 - accuracy: 0.9500 - val_loss: 0.0695 - val_accuracy: 1.0000
Epoch 19/20
7/7 [==============================] - 16s 2s/step - loss: 0.1445 - accuracy: 0.9275 - val_loss: 0.0645 - val_accuracy: 0.9688
Epoch 20/20
7/7 [==============================] - 16s 2s/step - loss: 0.0835 - accuracy: 0.9704 - val_loss: 0.0149 - val_accuracy: 1.0000
```

In [9]:
```
from tensorflow.keras.preprocessing import image
import numpy as np
img = image.load_img('79.jpg',target_size=(224,224))
img = image.img_to_array(img)
img = np.expand_dims(img,axis=0)   ### flattening
ypred = model.predict(img)
if ypred[0][0] == 1:
  print("Covid Negative")
else:
  print("Covid Positive")
```

```
Covid Positive
```

In [10]:
```
ypred[0][0]
```

Out[10]:
```
0.0
```

In [11]:
```
#### save the model
model.save("covid_model.h5")
```

In [12]:
```
import tensorflow.keras as tf
mymodel = tf.models.load_model("covid_model.h5")
```

- **COVID NEGATIVE:**

jupyter Day 5 - Covid Classification - CNN Last Checkpoint: a few seconds ago (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

```
7/7 [==============================] - 16s 2s/step - loss: 0.2119 - accuracy: 0.9110 - val_loss: 0.2436 - val_accuracy: 0.8750
Epoch 16/20
7/7 [==============================] - 17s 2s/step - loss: 0.2419 - accuracy: 0.8851 - val_loss: 0.1141 - val_accuracy: 0.9688
Epoch 17/20
7/7 [==============================] - 18s 3s/step - loss: 0.1875 - accuracy: 0.9294 - val_loss: 0.0757 - val_accuracy: 1.0000
Epoch 18/20
7/7 [==============================] - 17s 2s/step - loss: 0.1609 - accuracy: 0.9500 - val_loss: 0.0695 - val_accuracy: 1.0000
Epoch 19/20
7/7 [==============================] - 16s 2s/step - loss: 0.1445 - accuracy: 0.9275 - val_loss: 0.0645 - val_accuracy: 0.9688
Epoch 20/20
7/7 [==============================] - 16s 2s/step - loss: 0.0835 - accuracy: 0.9704 - val_loss: 0.0149 - val_accuracy: 1.0000
```

In [13]:
```
from tensorflow.keras.preprocessing import image
import numpy as np
img = image.load_img('80.jpg',target_size=(224,224))
img = image.img_to_array(img)
img = np.expand_dims(img,axis=0)   ### flattening
ypred = model.predict(img)
if ypred[0][0] == 1:
  print("Covid Negative")
else:
  print("Covid Positive")
```

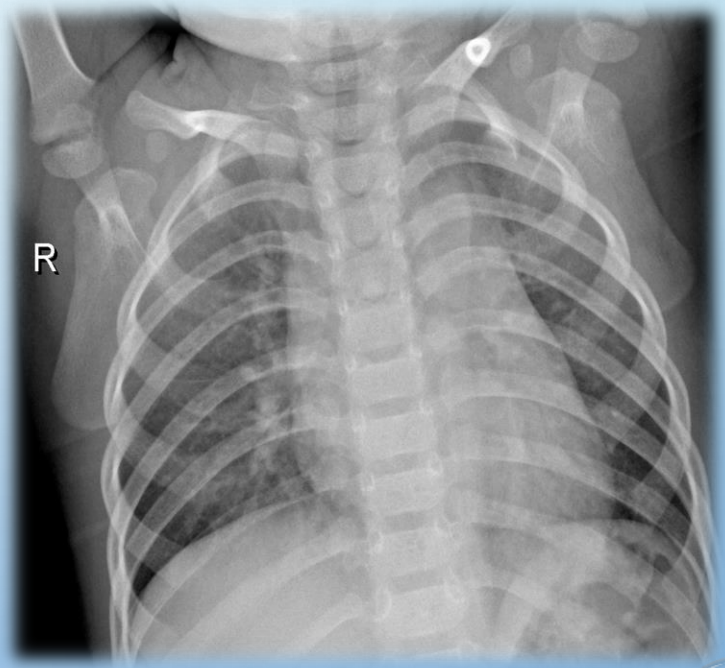
```
Covid Negative
```

In [10]:
```
ypred[0][0]
```

Out[10]:
```
0.0
```

In [11]:
```
#### save the model
model.save("covid_model.h5")
```

In [12]:
```
import tensorflow.keras as tf
mymodel = tf.models.load_model("covid_model.h5")
```



# PROJECT TIME LINE DIAGRAM

| S.NO | DATE | PLANNED WORK |
|---|---|---|
| 1 | 5/02/2021 - 6/02/2021 | TTCIE ABSTRACT SUBMISSION |
| 2 | 15/02/2021 - 20/02/2021 | TTCIE PAPER REVIEW SUBMISSION |
| 3 | 21/02/2021 - 25/02/2021 | MINI PROJECT CODE EXECUTION |
| 4 | 26/02/2021 - 28/02/2021 | PPT FOR MINI PROJECT |
| 5 | 1/03/2021 - 2/03/2021 | PROJECT REPORT |
| 6 | 3/03/2021 | FINISHING TOUCHES |

# REFERENCES:


1. https://www.google.com/amp/s/www.freecodecamp.org/news/an-intuitive-guide-to-convolutional-neural-networks-260c2de0a050/amp/
2. https://www.freecodecamp.org/news/an-intuitive-guide-to-convolutional-neural-networks-260c2de0a050/
3. https://www.superdatascience.com/blogs/convolutional-neural-networks-cnn-step-4-full-connection
4. Lauer SA, Grantz KH, Bi Q, Jones FK, Zheng Q, Meredith HR, Azman AS, Reich NG, Lessler J. The incubation period of coronavirus disease 2019 (COVID-19) from publicly reported confirmed cases: estimation and application. Ann Intern Med 2020;2019. https://doi.org/10.7326/M20-0504.
5. Jibril SIAML, Islam Md M, Sharif US. Predictive data mining models for novel coronavirus (COVID-19) infected patients recovery. SN Comput Sci 2020;1:206. https://doi.org/10.1007/s42979-020-00216-w.
6. Thevenot J, Lopez MB, Hadid A. A survey on computer vision for assistive medical diagnosis from faces. IEEE J Biomed Health Inf 2018;22:1497–511. https://doi.org/10.1109/JBHI.2017.2754861.