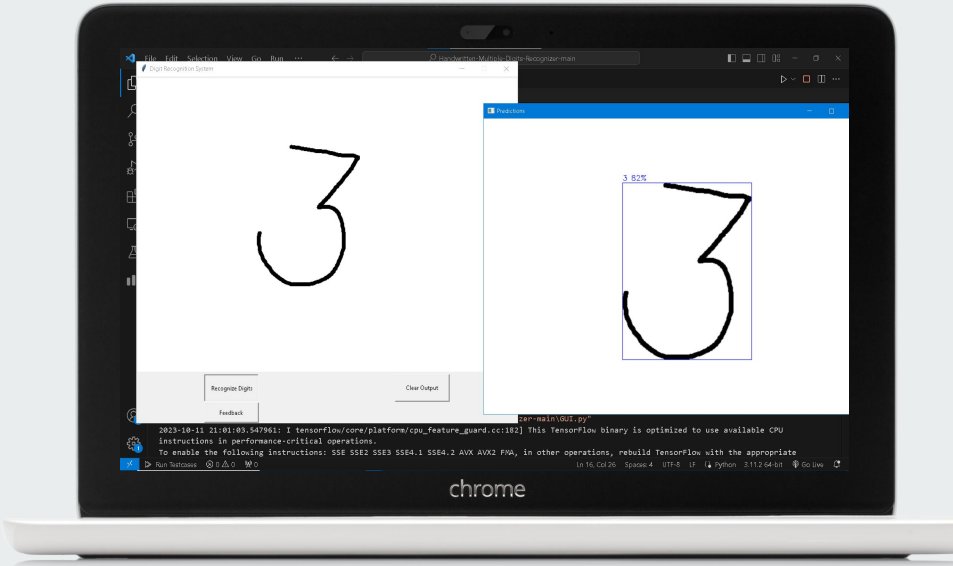
Minor Project Handwritten Digit recognition

Goal - Is to learn about the Deep Learning and Machine learning algorithms and implement a practical project.



INTRODUCTION?

Q1- What ?

Handwritten digit recognition can be used to automatically digitize large amounts of handwritten data, such as handwritten forms or documents, making it much easier to store and process the data

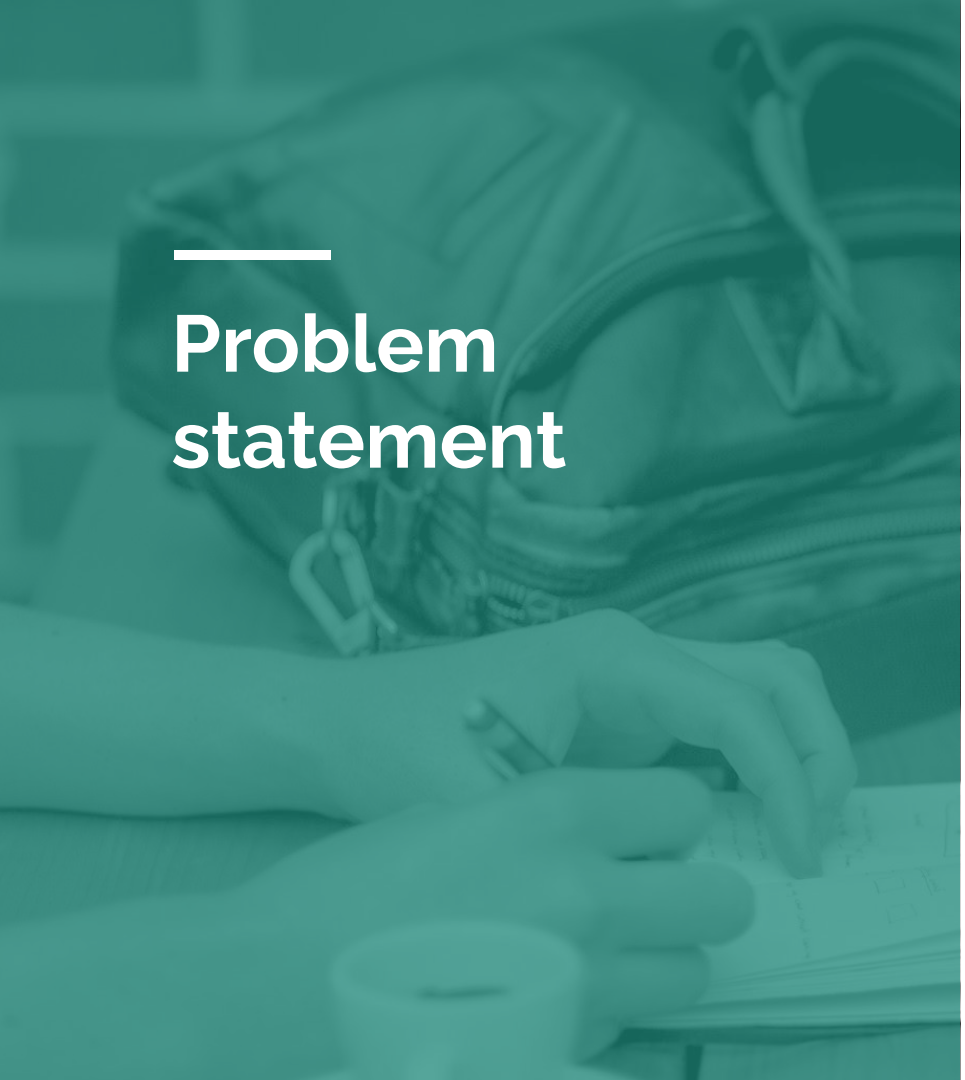
Q2-NEED ?

- Banking and finance
- Postal service
- Education
- Personalization

Q3 – IMPORTANCE?

.

- This can save time and reduce errors
- Improving efficiency:
- Enhancing accessibility



Problem statement

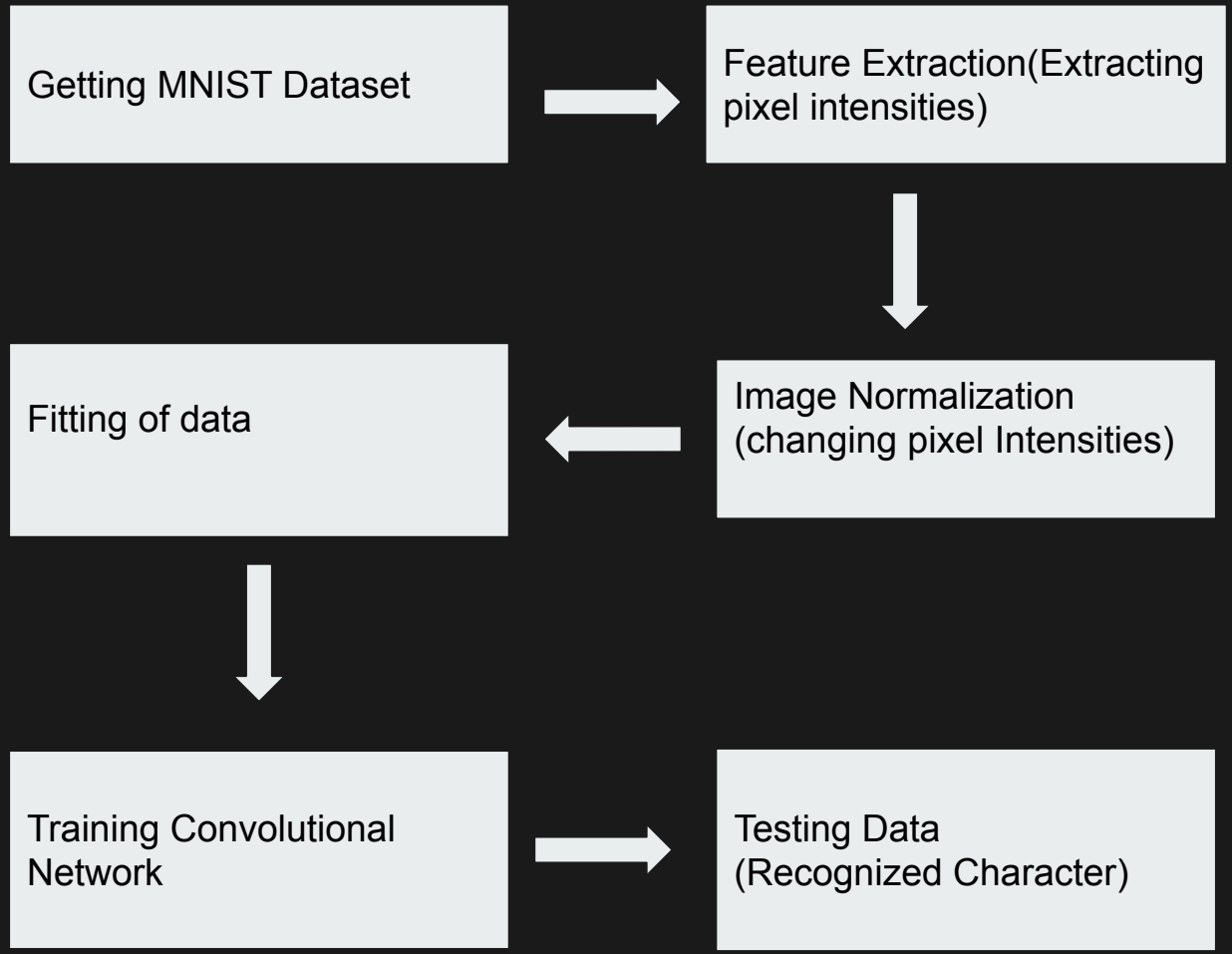
- One of the challenges in handwritten characters recognition wholly lies in the **variation and distortion of handwritten character** set because distinct community may use diverse style of handwriting, and control to draw the similar pattern of the characters of their recognized script.

- **Limited dataset size**: The availability of a large dataset is critical for training recognition systems. However, there are a limited number of labelled datasets available for handwritten digit recognition, which can limit the performance of the recognition system.

What are the algorithm available

Statistical approaches are established on planning of how data are selected. Algorithms utilizes the information of the statistical distribution of pixels in the image

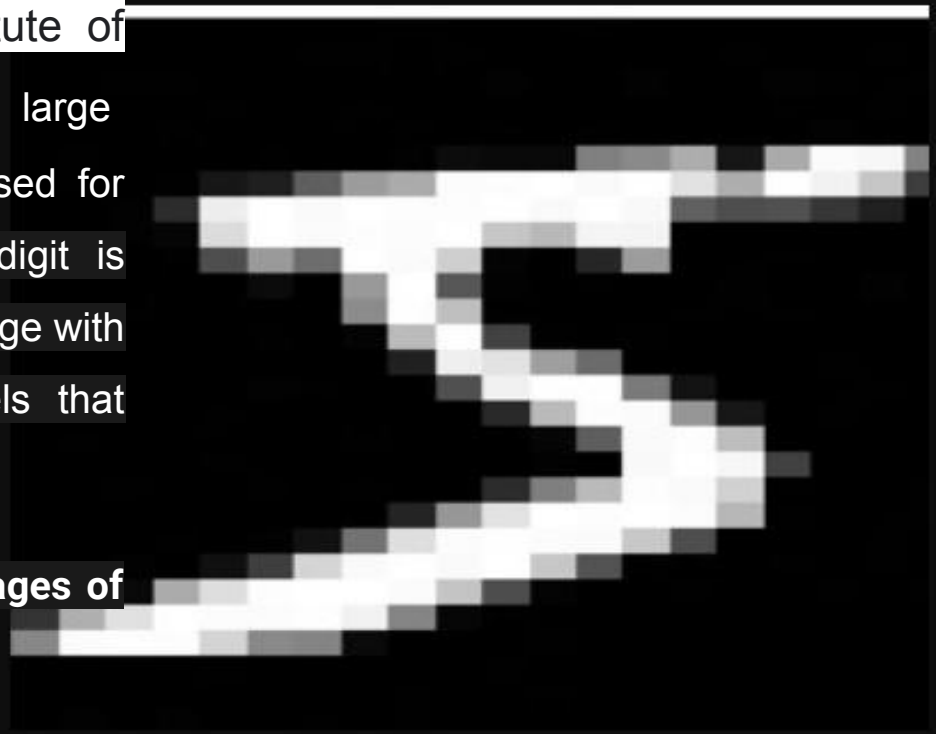
Several machines learning algorithm namely, **CNN,ANN, Support Vector Machine,Logistic regression, Random Forest and Decision** Tree has been used for the recognition of digits using MNIST DATASET.


Getting MNIST Dataset
Feature Extraction(Extracting pixel intensities)
Image Normalization (changing pixel Intensities)
Fitting of data
Training Convolutional Network
Testing Data (Recognized Character)

Dataset-MNIST

The MNIST database (Modified National Institute of Standards and Technology database) is a large database of handwritten digits that is commonly used for training various image processing systems.Each digit is normalized and centered in a gray-scale (0 - 255) image with size 28 × 28. Each image consists of 784 pixels that represent the features of the digits

MNIST contains a collection of 70,000, 28 x 28 images of handwritten digits from 0 to 9.



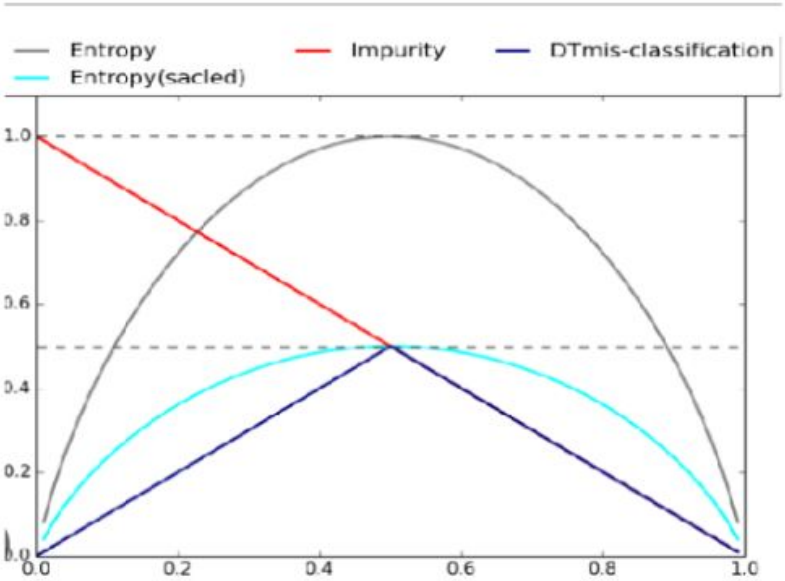
Results

Support Vector Machine

SNo	Kernel	C	F1 Score	Recall	Precision	Accuracy
1	linear	1	0.97768	0.977977	0.977873	0.977977
4	poly	1	0.9860745	0.986111	0.986196	0.98611
7	rbf	1	0.520404	0.4666666	0.91976	0.46666
2	linear	5	0.977686	0.977677	0.977873	0.977677
5	poly	5	0.98607	0.98611	0.986196	0.98611
8	rbf	5	0.5853035	0.522222	0.92114	0.5222222
3	linear	10	0.97768	0.977877	0.977873	0.977877
6	poly	10	0.986074	0.98611	0.9861964	0.9861111
9	rbf	10	0.58530	0.52222	0.921143	0.522222

Decision Tree 28X28

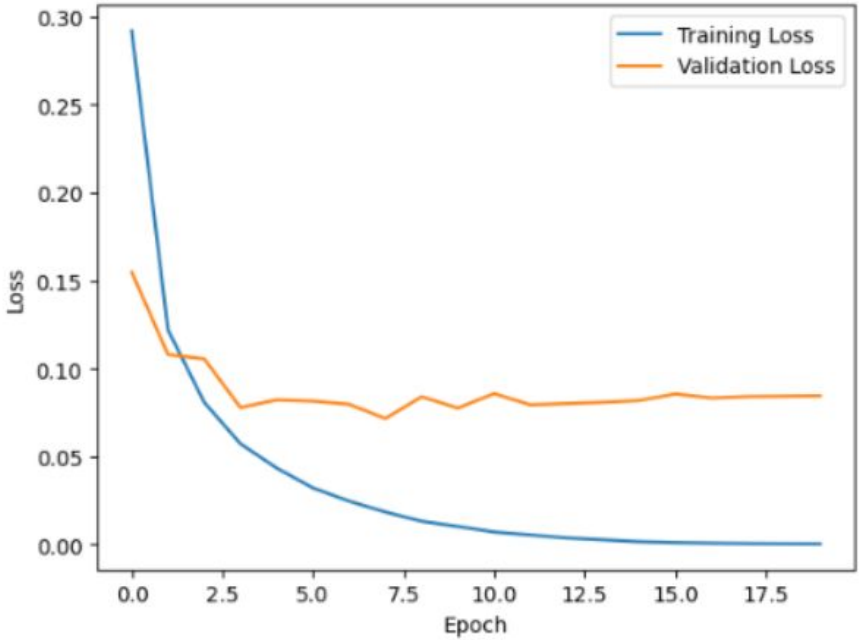
	CV=5	CV=10	CV=15
Accuracy	88%	85.65%	87.27%
Precision	88%	88.26%	85.65%
Recall	88%	85.02%	83.62%
FI-Score	88%	85.06%	88%

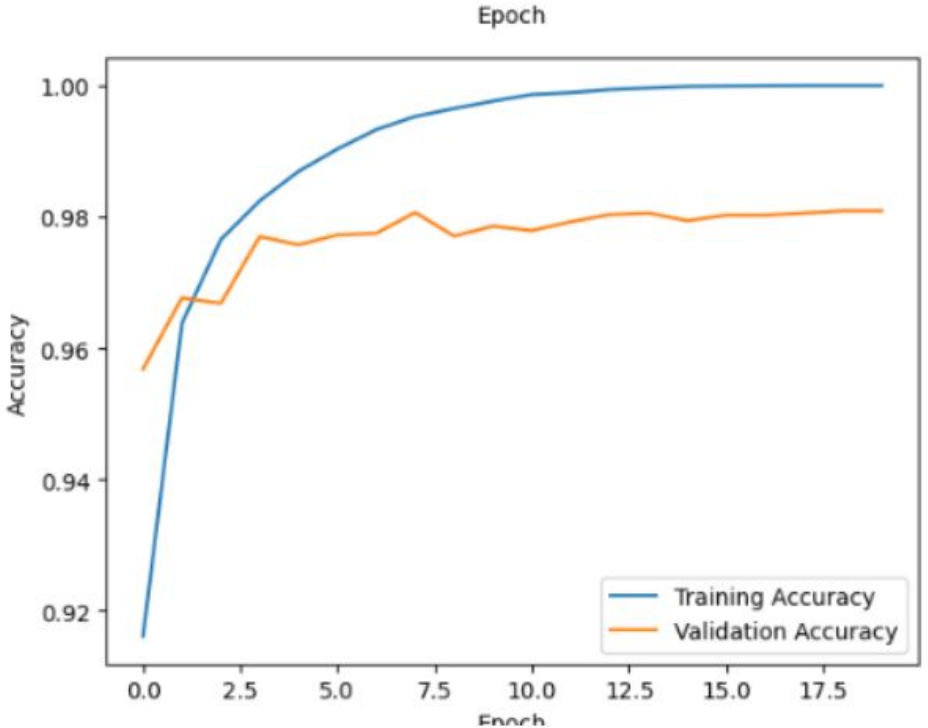


ANN 28*28

Activation	tanh	sigmoid	relu	tanh	relu	relu	sigmoid
Optimizer	adam	adam	adam	RMSprop	RMSprop	SGD	RMSprop
F1 - score	0.9787	0.9783	0.9790	0.9756	0.9786	0.9518	0.9762
Precision	0.97882	0.9784	0.9790	0.9757	0.9787	0.9518	0.9763
Recall	0.978	0.978	0.979	0.9756	0.9787	0.9519	0.9762
Accuracy	0.9788	0.9783	0.9788	0.9756	0.9786	0.9519	0.9761
Execution time	83.064 seconds	142.84 seconds	142.87 seconds	83.18198seconds	142.7178 seconds	62.910 seconds	80.0166261seconds

Here activation functions are referred to as the middle layers

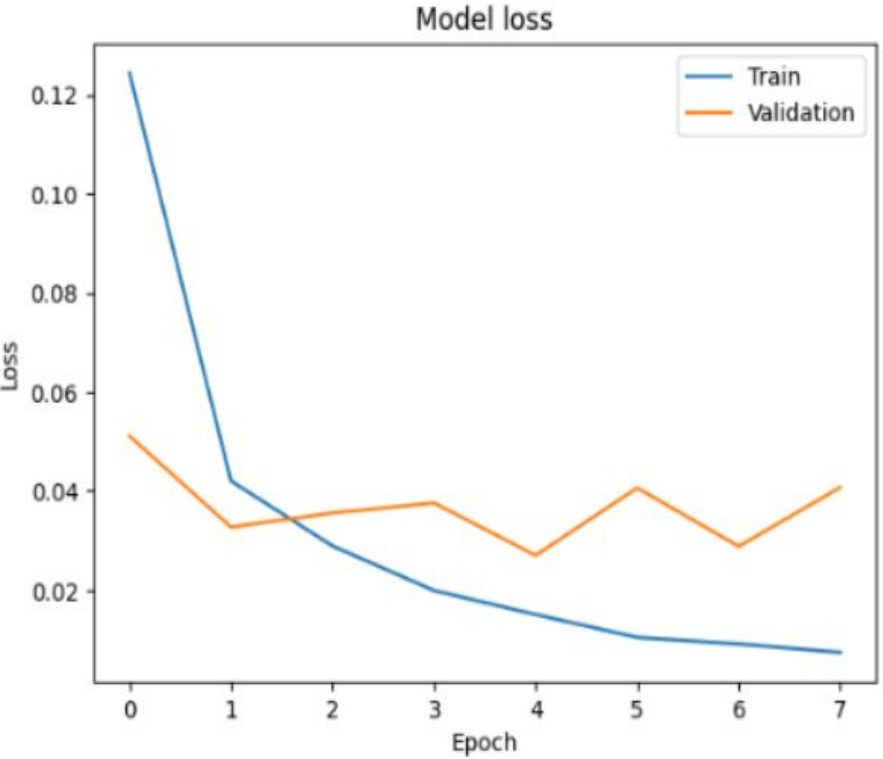

Training Loss
Validation Loss
Loss
Epoch
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5

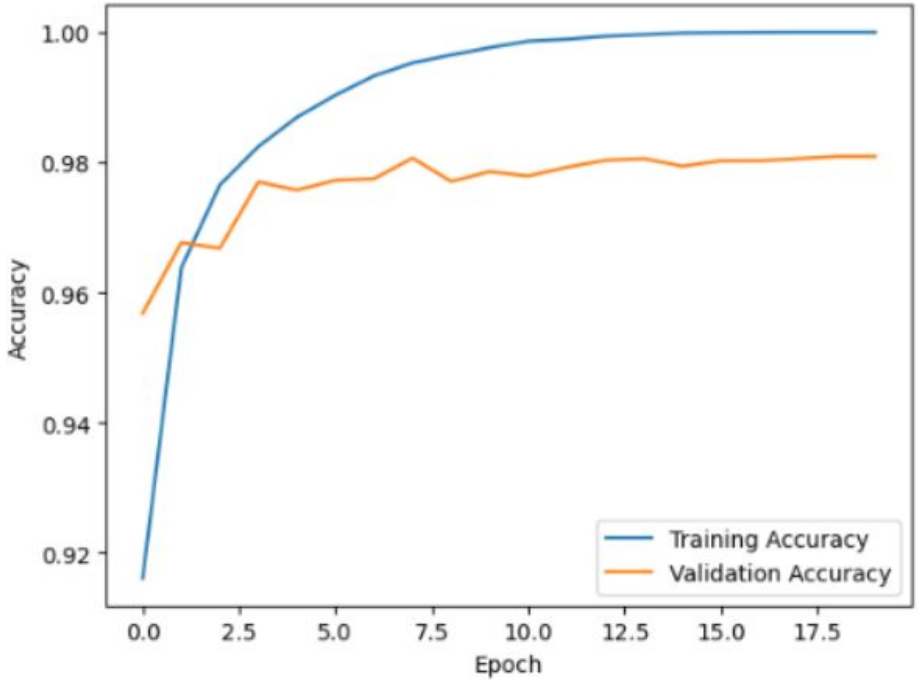

Epoch
Training Accuracy
Validation Accuracy
Accuracy
1.00
0.98
0.96
0.94
0.92
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5

CNN 28*28

Activation	tanh	sigmoid	relu	tanh	relu	relu	sigmoid
Optimizer	adam	adam	adam	RMSprop	RMSprop	SGD	RMSprop
F1 - score	0.9850	0.9829	0.9796	0.98677	0.9845	0.9759	0.9779
Precision	0.9850	0.9829	0.9803	0.9869	0.9849	0.9760	0.9781
Recall	0.9851	0.9830	0.9794	0.9866	0.9843	0.9760	0.9779
Accuracy	0.9850	0.9830	0.9797	0.9868	0.9846	0.9761	0.9781
Execution time	203.108 seconds	262.887 seconds	262.848 seconds	202.6773 seconds	262.75241 seconds	262.65 seconds	202.9133 seconds

Here activation functions are referred to as the middle layers


Model loss
Train
Validation
Loss
Epoch
0.12
0.10
0.08
0.06
0.04
0.02
0
1
2
3
4
5
6
7


Training Accuracy
Validation Accuracy
Accuracy
Epoch
1.00
0.98
0.96
0.94
0.92
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5

Inference

- CNN gives the best results than any other algo because CNN works well for image dataset as in this case, compared to ANN.
- Accuracy depends on many factors like trade of with time, resolution of image and hyper parameters
- If we reduce the features of image it greatly increase accuracy and reduce the time of execution
- Deep learning algorithm are far more better than Machine learning algorithms in image processing because of Ability to learn complex representations and Large-scale training.