

# HAND-WRITTEN RECOGNITION

---

Submitted To: Mitchell Harris

Submitted By : Ayush Jain

Subject: Fundamentals of Artificial Intelligence



---

# Number Plate Recognition

---

# Hand-Writing Recognition

---

# Student Work submitted online

# BUSINESS PROBLEM

---



Car Number Plate



Hand-Writing  
Recognition



Student Work  
submitted online

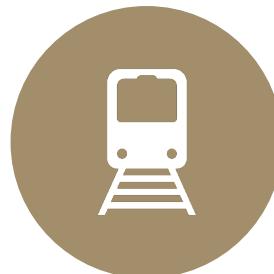


# ABOUT DATASET

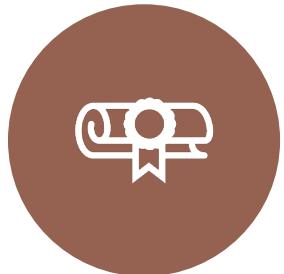
---



Source:  
<https://www.tensorflow.org/datasets/catalog/mnist>



Train set: 60000 Images  
Test set: 10000 Images



Array in greyscale RGB codes (from 0 to 255)  
labels from 0 to 9



Reshaping and Normalizing the Images



# WHAT DEFINE SUCCESS:

---



Accuracy: 99%



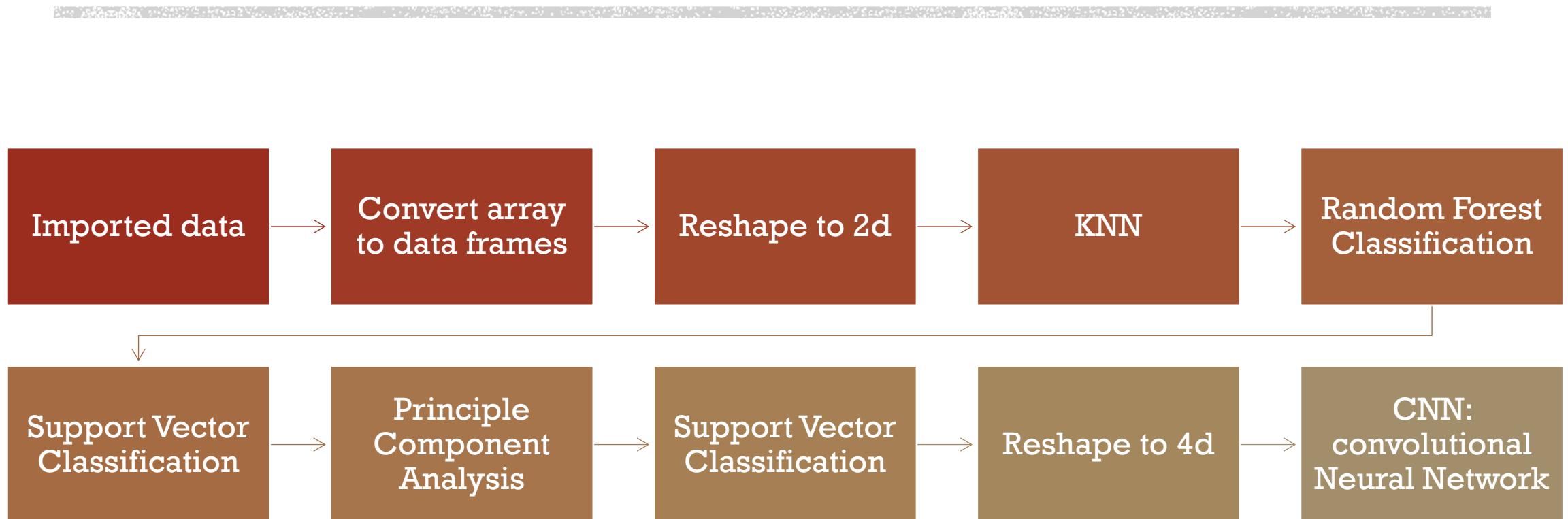
P-value: 1%



High Precision  
Required



# WORK-FLOW



# K-NEAREST NEIGHBORS

	precision	recall	f1-score	support
0	0.95	0.98	0.97	980
1	0.96	0.99	0.97	1135
2	0.96	0.93	0.94	1032
3	0.92	0.95	0.94	1010
4	0.94	0.94	0.94	982
5	0.93	0.92	0.93	892
6	0.96	0.97	0.97	958
7	0.94	0.92	0.93	1028
8	0.96	0.90	0.93	974
9	0.92	0.92	0.92	1009
accuracy			0.94	10000
macro avg	0.94	0.94	0.94	10000
weighted avg	0.94	0.94	0.94	10000

- Accuracy: 94%
- Reason of using: Algorithm doesn't learn instead works on the distance
- Work on respective category



# RANDOM FOREST CLASSIFIER

	precision	recall	f1-score	support
0	0.95	0.99	0.97	980
1	0.98	0.99	0.98	1135
2	0.94	0.96	0.95	1032
3	0.93	0.94	0.94	1010
4	0.95	0.95	0.95	982
5	0.94	0.93	0.93	892
6	0.96	0.96	0.96	958
7	0.96	0.94	0.95	1028
8	0.95	0.92	0.94	974
9	0.95	0.93	0.94	1009
accuracy			0.95	10000
macro avg	0.95	0.95	0.95	10000
weighted avg	0.95	0.95	0.95	10000

- Accuracy: 95%
- Reason: provide a feature importance estimate
- Estimates the test error without incurring
- Repeated model training with cross-validation



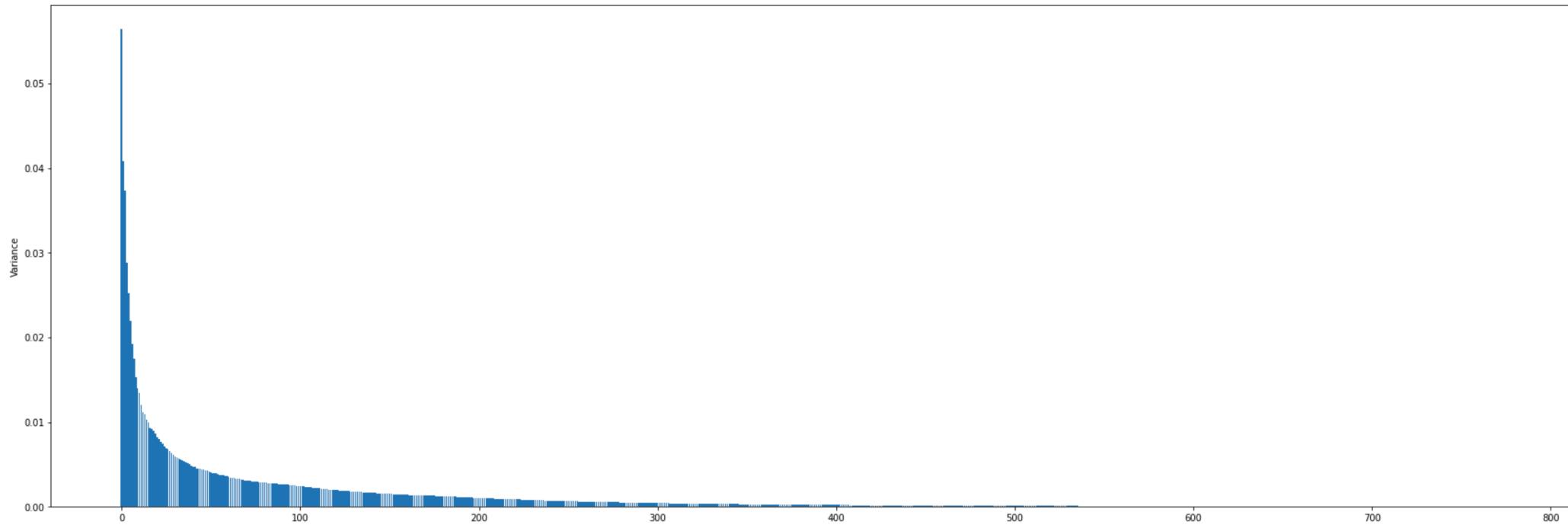
# SUPPORT VECTOR MACHINE

	precision	recall	f1-score	support
0	0.98	0.99	0.98	980
1	0.99	0.99	0.99	1135
2	0.96	0.97	0.96	1032
3	0.97	0.97	0.97	1010
4	0.97	0.96	0.97	982
5	0.96	0.96	0.96	892
6	0.98	0.97	0.98	958
7	0.93	0.96	0.94	1028
8	0.96	0.95	0.96	974
9	0.97	0.94	0.95	1009
accuracy			0.97	10000
macro avg	0.97	0.97	0.97	10000
weighted avg	0.97	0.97	0.97	10000

- Accuracy: 97%
- Clear margin of separation
- Effective in high dimensional spaces.



Text(0, 0.5, 'Variance')



# PRINCIPLE COMPONENT ANALYSIS

- Eigenvalues which sort components based on variance

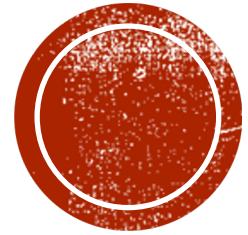


# SUPPORT VECTOR MACHINE

	precision	recall	f1-score	support
0	0.98	0.99	0.98	980
1	0.99	0.99	0.99	1135
2	0.97	0.96	0.96	1032
3	0.97	0.98	0.97	1010
4	0.97	0.97	0.97	982
5	0.96	0.97	0.96	892
6	0.98	0.97	0.98	958
7	0.94	0.96	0.95	1028
8	0.96	0.95	0.96	974
9	0.97	0.95	0.96	1009
accuracy			0.97	10000
macro avg	0.97	0.97	0.97	10000
weighted avg	0.97	0.97	0.97	10000

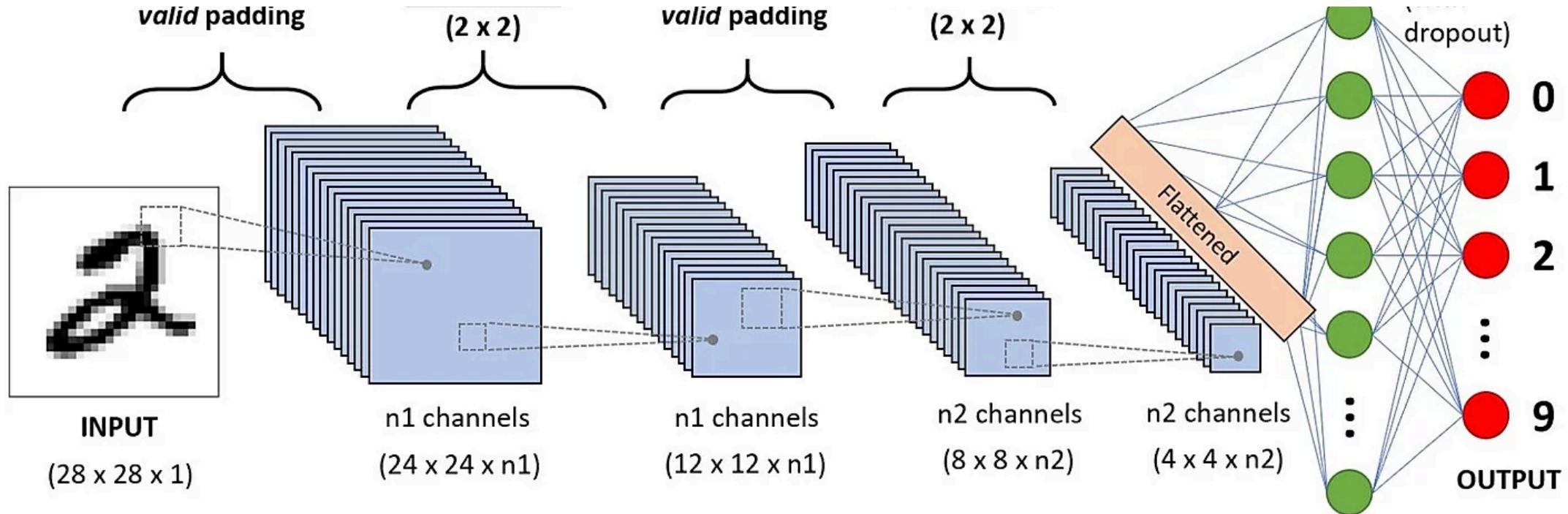
- Only 2 & 7 has improvement of 1 %
- Rest no significance change





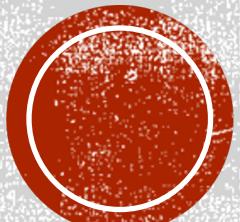
# CNN: CONVOLUTIONAL NEURAL NETWORK





## HOW & WHY CNN?

- Spatial and Temporal dependencies the application of relevant filters.
- The architecture performs a better fitting to the image dataset
- Due to the reduction in the number of parameters involved
- The network can be trained to understand the sophistication of the image better.

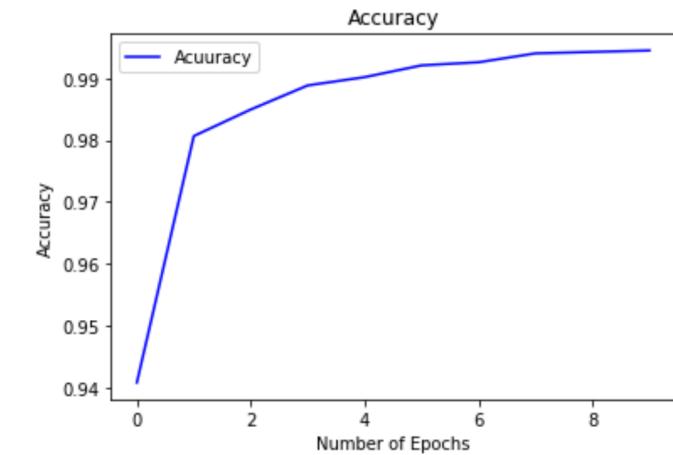
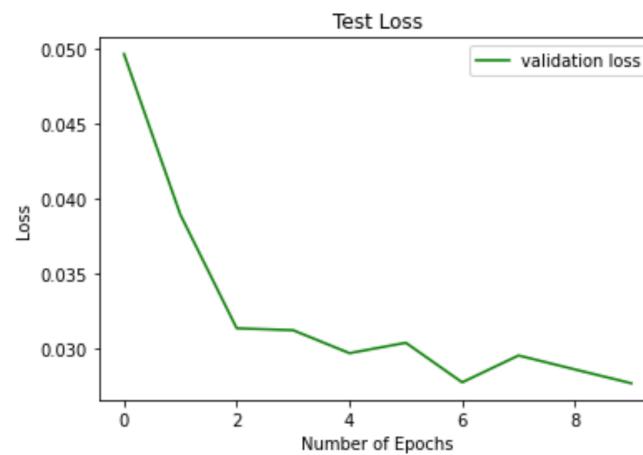
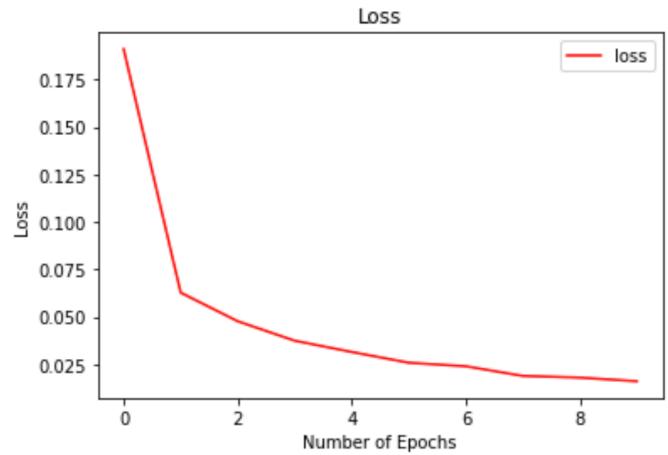


```
Train on 60000 samples, validate on 10000 samples
Epoch 1/10
60000/60000 [=====] - 143s 2ms/step - loss: 0.1906 - accuracy: 0.9420 - val_loss: 0.0726 - val_accuracy: 0.9761
Epoch 2/10
60000/60000 [=====] - 143s 2ms/step - loss: 0.0642 - accuracy: 0.9807 - val_loss: 0.0343 - val_accuracy: 0.9885
Epoch 3/10
60000/60000 [=====] - 147s 2ms/step - loss: 0.0455 - accuracy: 0.9860 - val_loss: 0.0325 - val_accuracy: 0.9892
Epoch 4/10
60000/60000 [=====] - 143s 2ms/step - loss: 0.0365 - accuracy: 0.9888 - val_loss: 0.0311 - val_accuracy: 0.9893
Epoch 5/10
60000/60000 [=====] - 143s 2ms/step - loss: 0.0309 - accuracy: 0.9907 - val_loss: 0.0341 - val_accuracy: 0.9894
Epoch 6/10
60000/60000 [=====] - 143s 2ms/step - loss: 0.0250 - accuracy: 0.9919 - val_loss: 0.0291 - val_accuracy: 0.9899
Epoch 7/10
60000/60000 [=====] - 147s 2ms/step - loss: 0.0209 - accuracy: 0.9935 - val_loss: 0.0290 - val_accuracy: 0.9902
Epoch 8/10
60000/60000 [=====] - 144s 2ms/step - loss: 0.0189 - accuracy: 0.9942 - val_loss: 0.0318 - val_accuracy: 0.9903
Epoch 9/10
60000/60000 [=====] - 143s 2ms/step - loss: 0.0161 - accuracy: 0.9949 - val_loss: 0.0306 - val_accuracy: 0.9904
Epoch 10/10
60000/60000 [=====] - 143s 2ms/step - loss: 0.0144 - accuracy: 0.9956 - val_loss: 0.0312 - val_accuracy: 0.9908
```

---

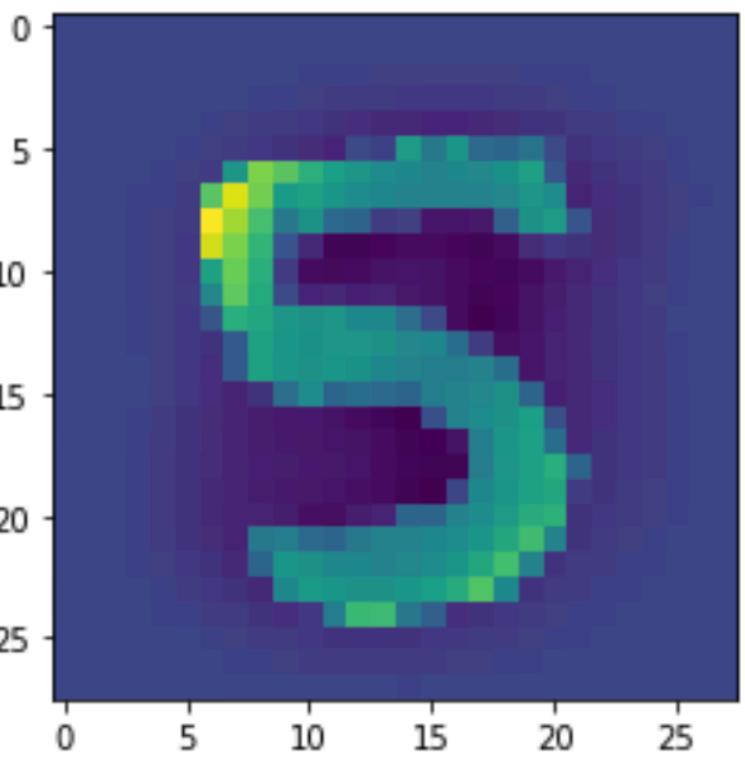
# RESULTS AFTER CNN





# GRAPH FOR LOSS & IMPROVEMENT



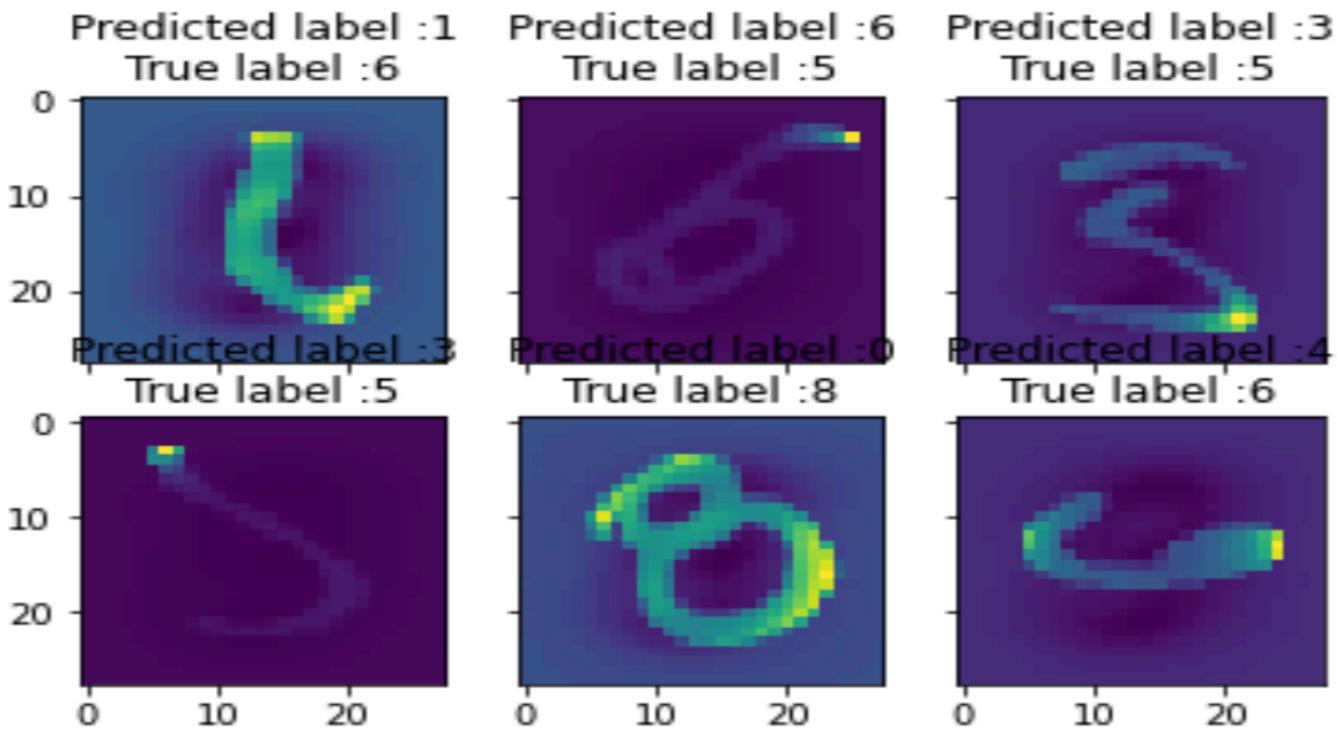


The predicted number is: 5

## TEST RESULTS

- Test Image
- Test label





---

# CASES OF FAILURE



# REFERENCES

- <https://www.analyticsvidhya.com/blog/2018/12/guide-convolutional-neural-network-cnn/>
- <https://towardsdatascience.com/dimensionality-reduction-does-pca-really-improve-classification-outcome-6e9ba21f0a32>
- <https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/>
- <https://www.thelearningmachine.ai/cnn>

