Great Learning
POWER AHEAD

SVHN DIGIT RECOGNITION

DEEP LEARNING PROJECT

MIT Applied Data Science Program
Elective Project: Deep learning
Delia McNamara
29 July 2024

CONTENTS / AGENDA

→ Business Problem Overview and Solution Approach

→ Data Preprocessing for ANNs

→ Model Performance Summary for ANNs

→ Data Preprocessing for CNNs

→ Model Performance Summary for CNNs

→ Conclusion

BUSINESS PROBLEM OVERVIEW & SOLUTION APPROACH

→ Seeking to create a neural network model to identify numerical digits in images.

→ SVHN dataset utilized for model training & testing.

- ◊ Pre-labeled images obtained from Google Street View containing numerical digits.
- ◊ Images cropped prior to import to contain only digits.

→ Neural network created will be able to identify digits contained within images without any further external image processing.

→ Artificial neural network (ANN) modeling will be performed for initial modeling.

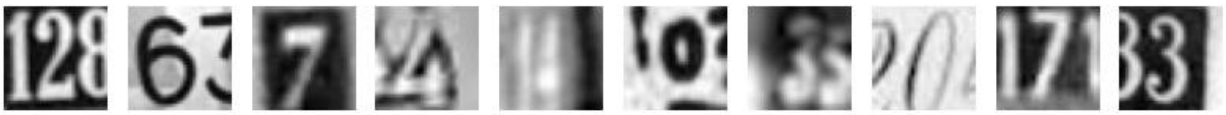
→ Convolutional neural network (CNN) modeling will be performed to determine the suitability of convolution for this dataset.

→ The best model of all ANNs & CNNs created will be recommended for use in digit identification.

DATA PREPROCESSING FOR ANNs

DATA OVERVIEW

- **60,000 labeled images from Street View House Numbers (SVHN) dataset.**

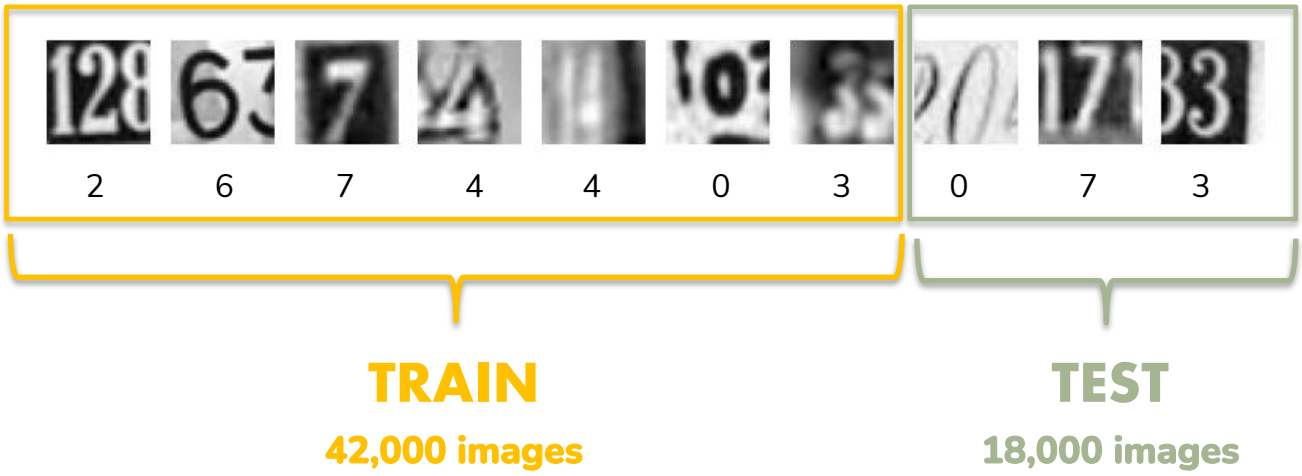


Label for each of the above images: 2, 6, 7, 4, 4, 0, 3, 0, 7, 3.

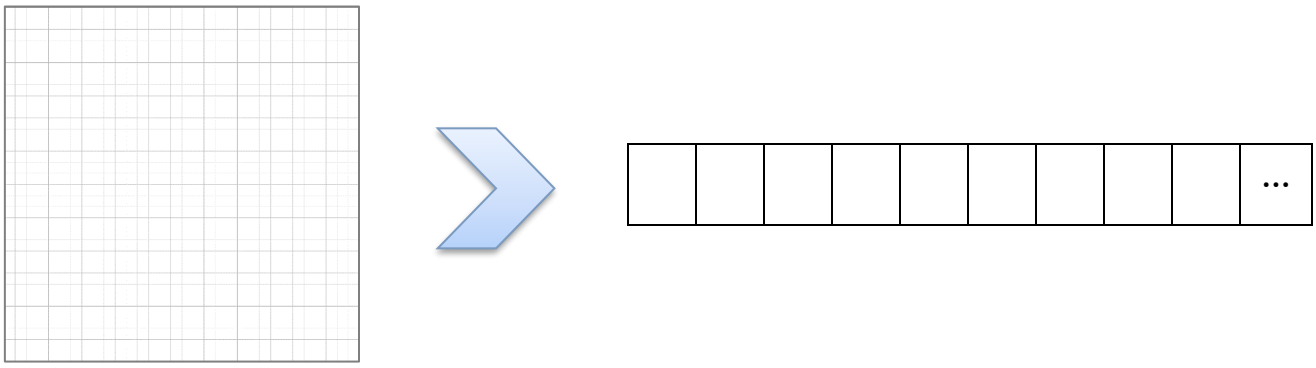
Example images from SVHN dataset. Images include digits 0-9.

Metadatum	Value
Image size (px)	32 x 32
Color channels	1 (grayscale)
No. images	60,000

ANN PREPROCESSING



- Dataset pre-split for model training & validation.
  - 42,000 images to train (70%).
  - 18,000 images to test (30%).



- Flattened independent variable 32 x 32 px images into 1D arrays.
  - Length 1,024 channel values per image.

NON -NORMALIZED

33.0704	30.2601	26.852	...	49.6682	50.853	53.0377

NORMALIZED

0.12969	0.11867	0.1053	...	0.19478	0.1994	0.20799

- One-hot encoded dependent variable for modeling.

- Normalized independent variable to values between 0 and 1.
  - Previously 0-255.

Y
3
4
1
0
8

0	1	2	3	4	5	6	7	8	9
0	0	0	1	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0

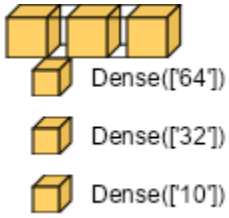
- One-hot encoded dependent variable for modeling.

NEW DATASET SHAPE

Variable	Dimension	Attributes
X (train)	42,000 x 1,024	Normalized 1D array
X (test)	18,000 x 1,024	Normalized 1D array
Y (train)	42,000 x 10	One-hot encoded vector
Y (test)	18,000 x 10	One-hot encoded vector

MODEL PERFORMANCE SUMMARY FOR ANNs

ANN MODEL 1



→ ANN model with:

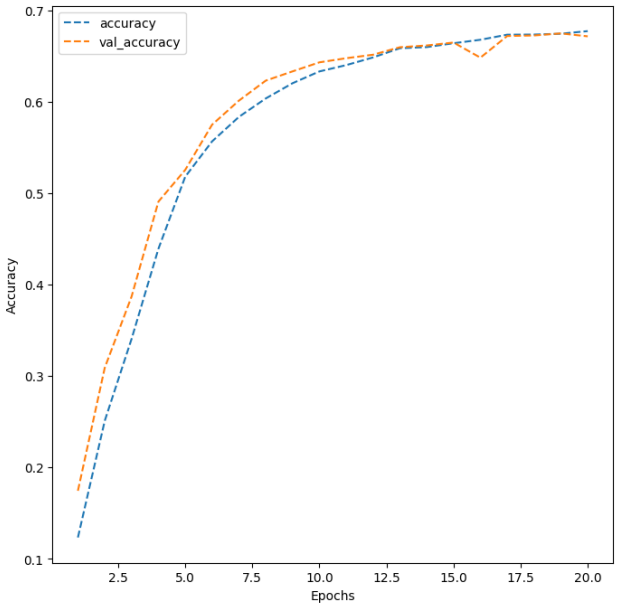
- ◊ 2 dense hidden layers.
  - ■ ReLU activation.
- ◊ 1 output layer.
  - ■ 10 nodes.
  - ■ Softmax

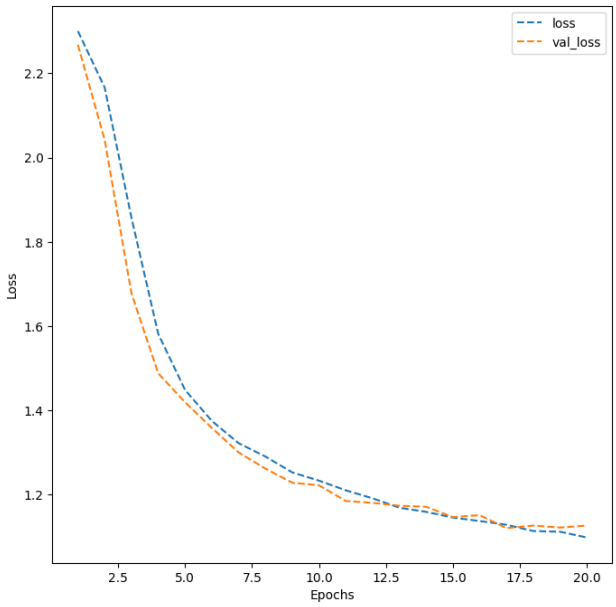
→ Compiled with:

- ◊ Optimizer: Adam
  - ■ Learning rate: 0.001
- ◊ Loss: Categorical cross-entropy

→ Fitted with:

- ◊ Validation split: 20%
- ◊ Batch size: 128
- ◊ 20 epochs





→ **Accuracy roughly 67%.**

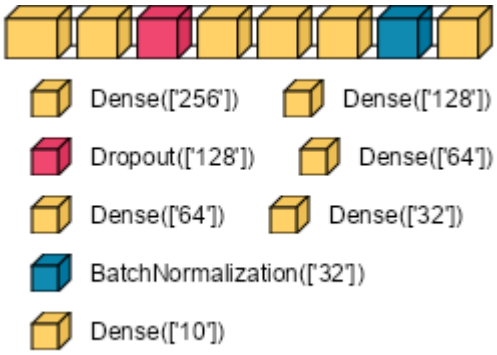
→ **Loss roughly 1.10.**

→ **Very little overfitting.**

→ **Good learning rate.**

→ **Seek to improve accuracy.**

ANN MODEL 2



→ ANN model with:

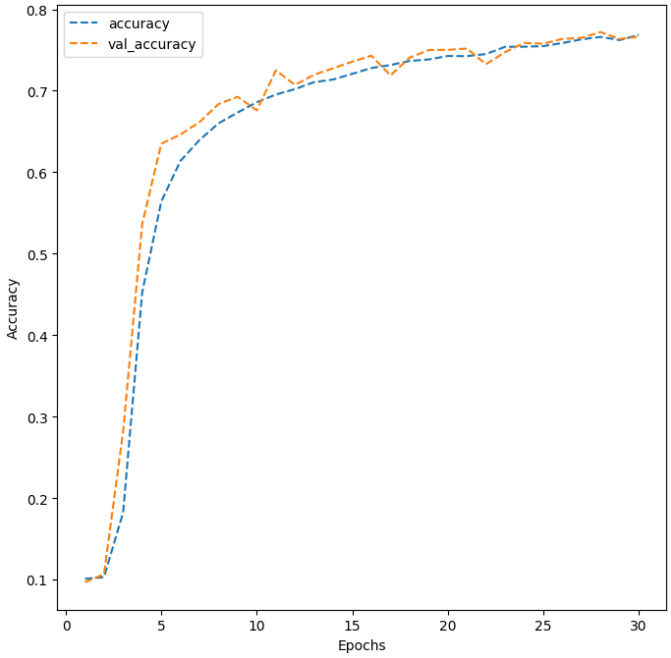
- ◊ 5 dense hidden layers.
  - ■ ReLU activation.
- ◊ 1 dropout layer.
- ◊ 1 normalization layer.
- ◊ 1 output layer.
  - ■ 10 nodes.
  - ■ Softmax activation.

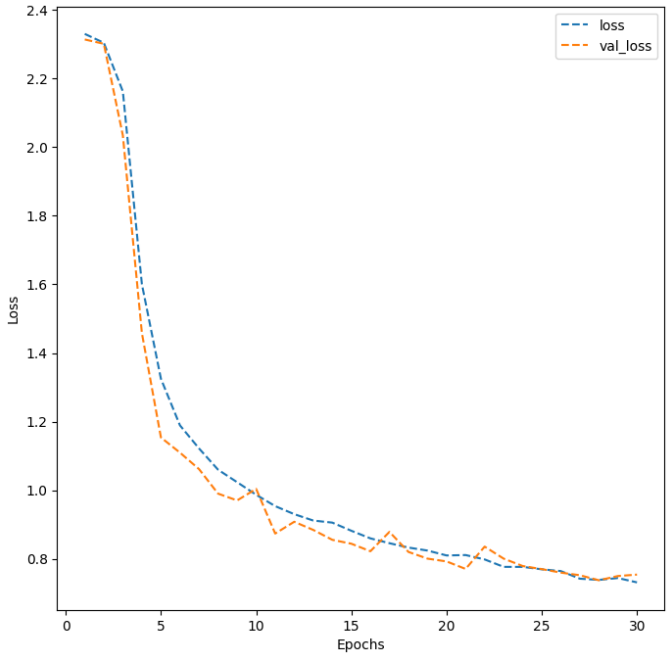
→ Compiled with:

- ◊ Optimizer: Adam
  - ■ Learning rate: 0.0005
- ◊ Loss: Categorical cross-entropy

→ Fitted with:

- ◊ Validation split: 20%
- ◊ Batch size: 128
- ◊ 30 epochs





→ Accuracy roughly 77%.

◊ Improvement from previous model.

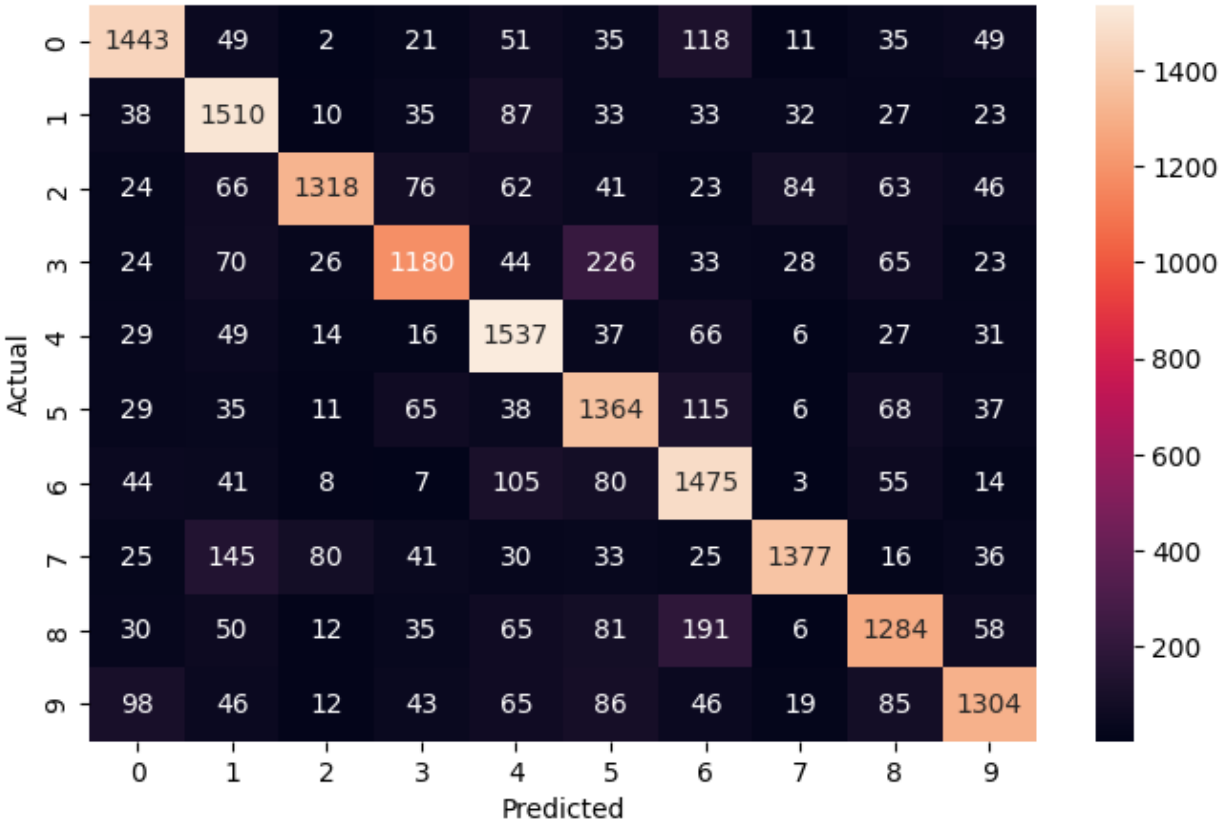
→ Loss roughly 0.73.

◊ Improvement from previous model.

→ Very little overfitting.

→ Slightly high learning rate.

→ Seek to improve accuracy.



→ **Fewest correctly predicted 3s**

- ◊ **7s also low for correct predictions.**

→ **3s predicted frequently as 5s.**

→ **Many high values for all digits predicted to actual.**

- ◊ **All digits predicted incorrectly many times.**

	Precision (%)	Recall (%)	F1 Score (%)	Support (%)
0	81	80	80	1,814
1	73	83	78	1,828
2	88	73	80	1,803
3	78	69	73	1,719
4	74	85	79	1,812
5	68	77	72	1,768
6	69	81	75	1,832
7	88	76	81	1,808
8	74	71	73	1,812
9	80	72	76	1,804
Accuracy			77	18,000
Total avg.	77	77	77	18,000
Weighted avg.	77	77	77	18,000

→ **Metrics range from 68-88%.**

- ◊ **No digits are being predicted well.**

→ **Digit 3 recall lowest.**

- ◊ **Model is predicting true negatives ok, but true positives are lower in comparison to other digits.**

→ **Digit 5 precision lowest.**

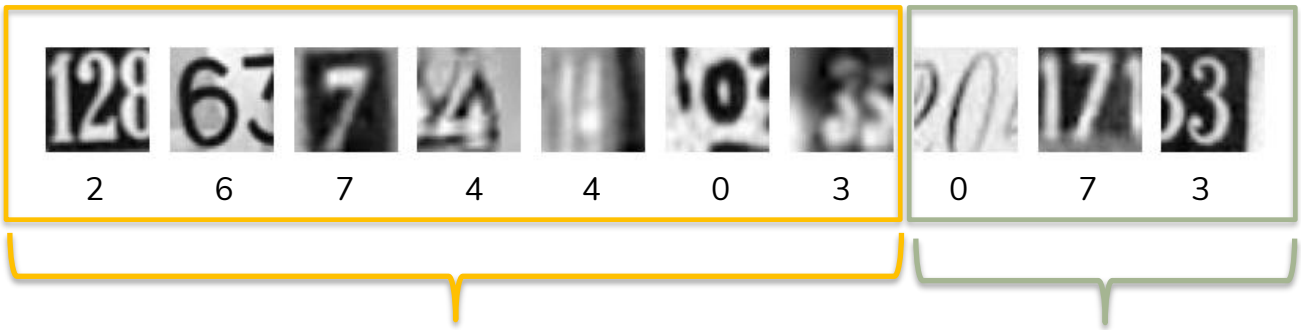
- ◊ **High proportion of false positives to true positives.**

→ **F1 scores all 72% or higher.**

- ◊ **Precision and recall are fairly well-balanced for all digits.**

DATA PREPROCESSING FOR CNNs

CNN PREPROCESSING



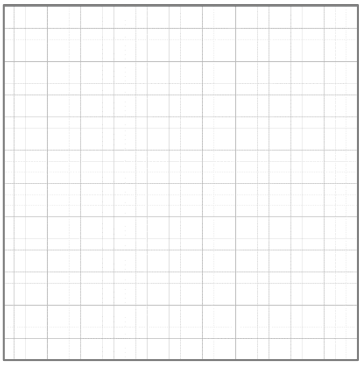
TRAIN

42,000 images

TEST

18,000 images

- **Dataset pre-split for model training & validation.**
  - **42,000 images to train (70%).**
  - **18,000 images to test (30%).**



Batch size	Height	Width	Depth
0	32	32	1

- Reshaped independent variable 32 x 32 px images into 4D arrays.

NON -NORMALIZED

33.0704	30.2601	26.852	...	49.6682	50.853	53.0377

NORMALIZED

0.12969	0.11867	0.1053	...	0.19478	0.1994	0.20799

- Normalized independent variable to values between 0 and 1.
  - Previously 0-255.

Y
3
4
1
0
8

0	1	2	3	4	5	6	7	8	9
0	0	0	1	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0

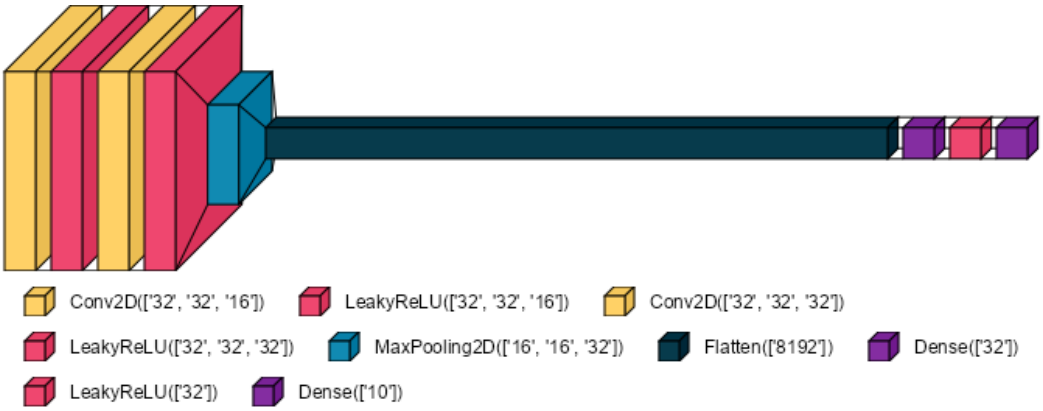
- **One-hot encoded dependent variable for modeling.**

NEW DATASET SHAPE

Variable	Dimension	Attributes
X (train)	42,000 x 32 x 32 x 1	Normalized 4D array
X (test)	18,000 x 32 x 32 x 1	Normalized 4D array
Y (train)	42,000 x 10	One-hot encoded vector
Y (test)	18,000 x 10	One-hot encoded vector

MODEL PERFORMANCE SUMMARY FOR CNNs

CNN MODEL 1



→ CNN model with:

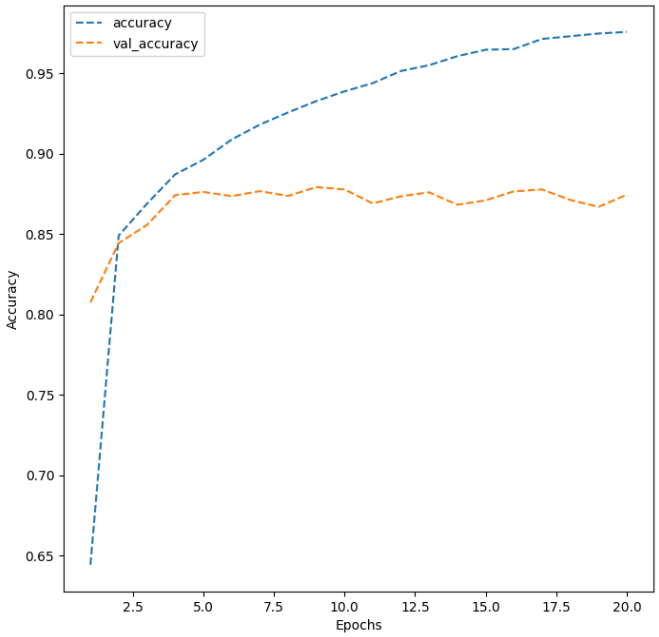
- ◊ 2 convolutional layers.
  - Output size same as input.
  - Kernel size 3 x 3.
- ◊ 3 Leaky ReLU layers.
  - Slope of 0.1.
- ◊ 1 max-pooling layer.
  - Pool size 2 x 2.
- ◊ 1 flatten layer.
- ◊ 1 dense layer.
- ◊ 1 output layer.
  - 10 nodes.
  - Softmax activation.

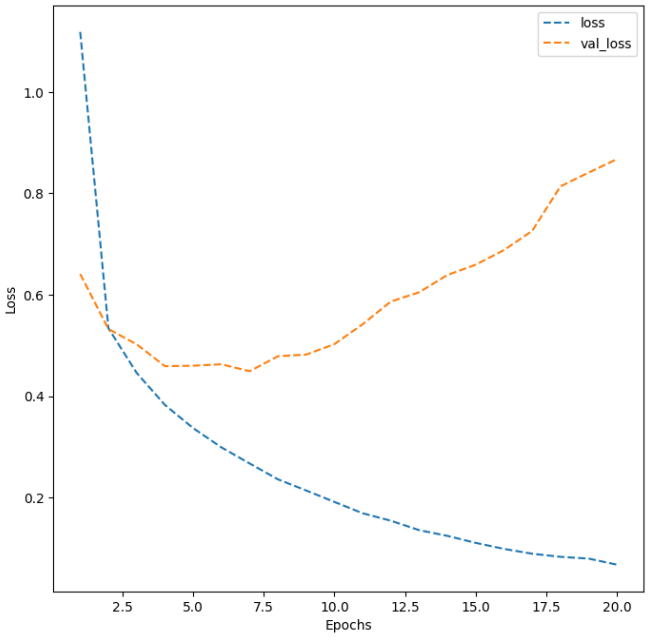
→ Compiled with:

- ◊ Loss: categorical cross-entropy.
- ◊ Optimizer: Adam.
  - Learning rate: 0.001.

→ Fitted with:

- ◊ Validation split: 20%.
- ◊ Batch size: 32.
- ◊ 20 epochs.





→ **Accuracy roughly 98%.**

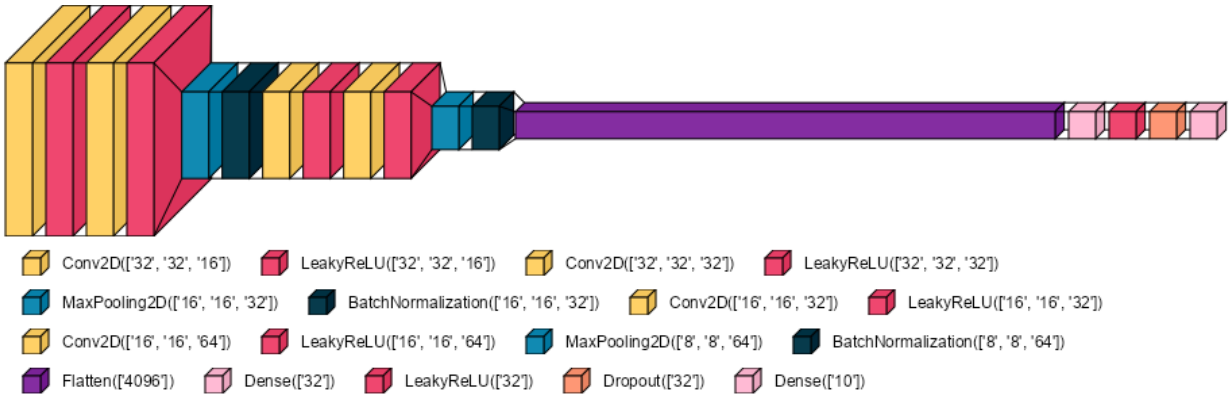
→ **Loss roughly 0.07.**

→ **Very overfit.**

→ **Prediction error increasing after 7.5 epochs.**

→ **Seek to reduce overfitting.**

CNN MODEL 2



→ CNN model with:

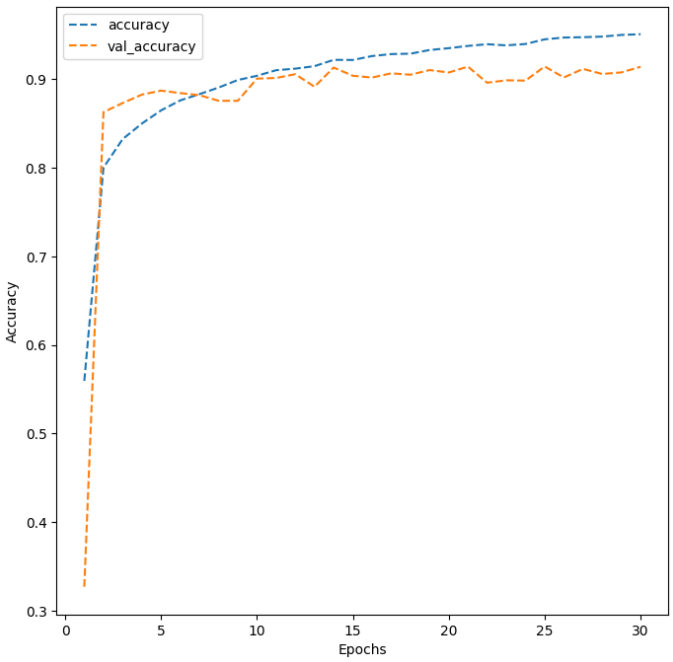
- ◊ 4 convolutional layers.
  - ○ Output size same as input.
  - ○ Kernel size 3 x 3.
- ◊ 5 Leaky ReLU layers.
  - ○ Slope of 0.1.
- ◊ 2 max-pooling layers.
  - ○ Pool size 2 x 2.
- ◊ 2 normalization layers.
- ◊ 1 flatten layer.
- ◊ 1 dense layer.
- ◊ 1 dropout layer.
- ◊ Rate of 0.5.
- ◊ 1 output layer.
  - ○ 10 nodes.
  - ○ Softmax activation.

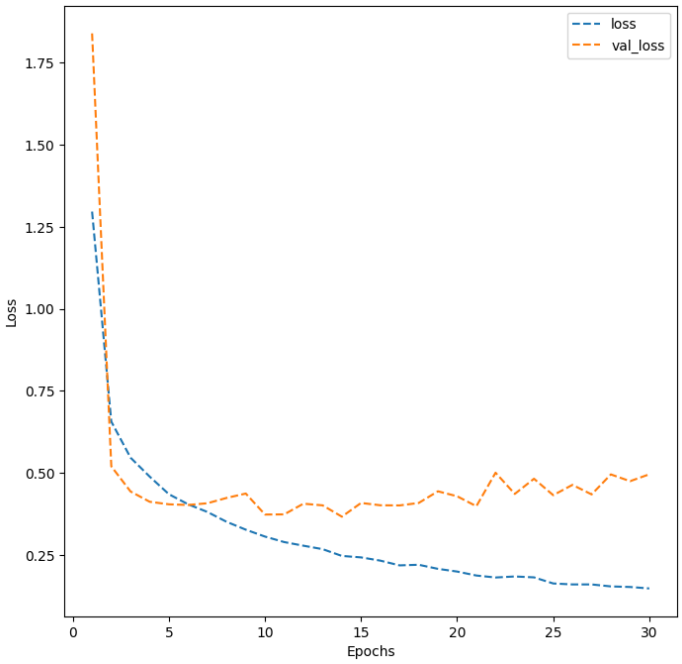
→ Compiled with:

- ◊ Loss: categorical cross-entropy.
- ◊ Optimizer: Adam.
  - ○ Learning rate: 0.001.

→ Fitted with:

- ◊ Validation split: 20%.
- ◊ Batch size: 128.
- ◊ 30 epochs.





→ Accuracy roughly 95%.

- ◊ Less than previous CNN.

→ Loss roughly 0.15.

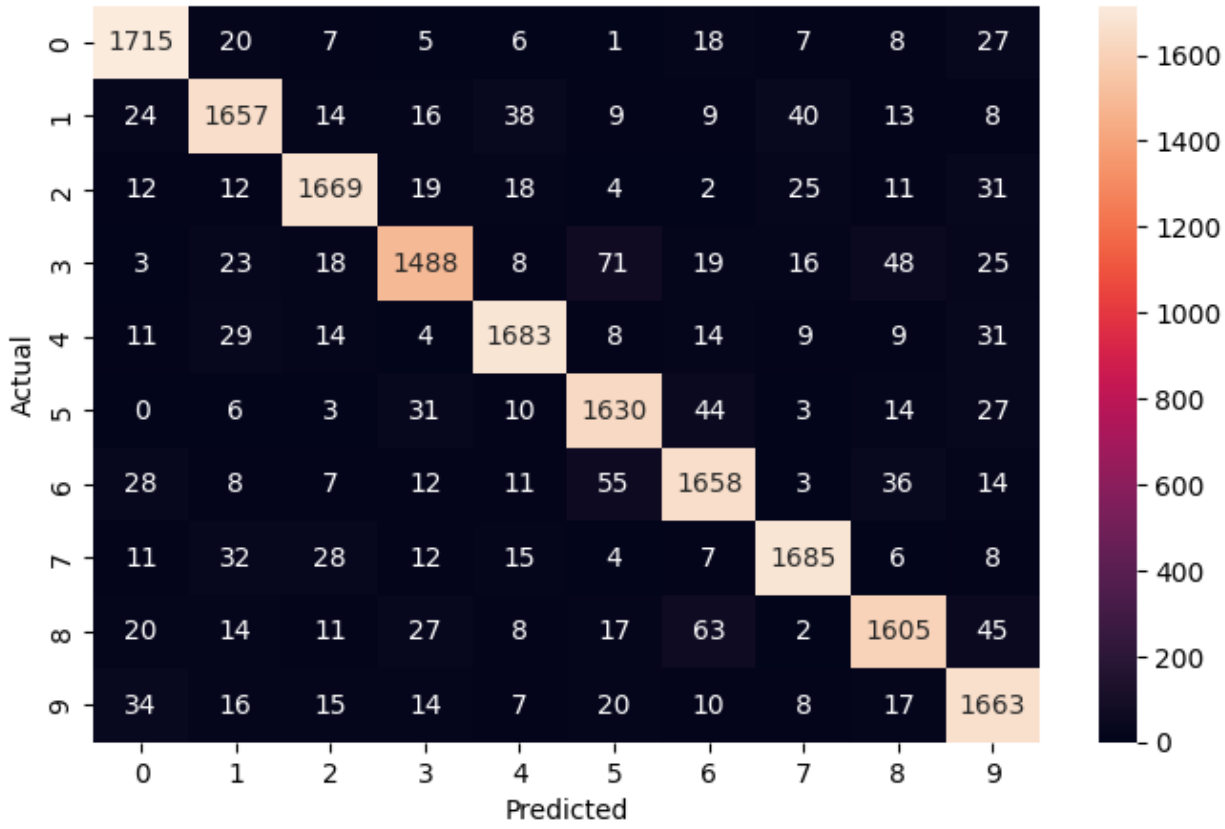
- ◊ Higher than previous CNN.

→ Slight overfitting.

- ◊ Significantly less overfitting than previous CNN
- ◊ Accuracy & loss decrease acceptable for better fit.

→ Prediction error increasing very slightly after 19 epochs.

→ Best performance of all models for accuracy without overfitting.



→ **High values for true positives.**

- ◊ **Many good predictions.**

→ **Lower values for incorrect predictions in comparison to ANN.**

- ◊ **Significant improvement.**

	Precision (%)	Recall (%)	F1 Score (%)	Support (%)
0	92	95	93	1,814
1	91	91	91	1,828
2	93	93	93	1,803
3	91	87	89	1,719
4	93	93	93	1,812
5	90	92	91	1,768
6	90	91	90	1,832
7	94	93	93	1,808
8	91	89	90	1,812
9	89	92	90	1,804
Accuracy			91	18,000
Total avg.	91	91	91	18,000
Weighted avg.	91	91	91	18,000

→ **All metrics over 85% for all digits.**

- ◊ **Great improvement in comparison to ANN.**

→ **Digit 3 recall lowest.**

- ◊ **Model is predicting true negatives well, but true positives are lower in comparison to other digits.**

→ **Precision scores all 89% or higher.**

- ◊ **High proportion of true positives to false positives.**

→ **F1 scores all 89% or higher.**

- ◊ **Precision and recall are well-balanced for all digits.**

CONCLUSIONS

→ **Convolution improved model fit well.**

→ **ANNs were less prone to overfitting but showed low accuracy, precision, and recall in comparison to CNNs.**

→ **CNN model 2 showed the best accuracy, precision, & recall with the least overfitting.**

- ◊ **Recommend CNN model 2 for digit recognition.**