# NN

## Model 1(Simple without hidden layers):

## Description:

The first layer is a **Flatten** layer, which takes an input with a shape of (100, 100, 1) and flattens it into a 1D array of size 10000. The input shape of (100, 100, 1) corresponds to 100x100 images with a single channel (e.g. grayscale).

The second layer is a **Dense** layer with 10 units and a sigmoid activation function.

Calculating MSE and MAE of Model 1

MSE: 19.83

MAE: 3.80

```
Classification Report
             precision    recall  f1-score   support

           0       0.53      0.89      0.67         9
           1       0.86      0.75      0.80         8
           2       1.00      0.33      0.50         9
           3       0.83      0.56      0.67         9
           4       0.43      0.60      0.50        10
           5       0.78      0.88      0.82         8
           6       0.50      0.44      0.47         9
           7       0.56      0.62      0.59         8
           8       0.72      0.72      0.72        18
           9       0.82      0.75      0.78        12

    accuracy                           0.66       100
   macro avg       0.70      0.65      0.65       100
weighted avg       0.70      0.66      0.66       100
```

loss: 0.9861 - accuracy: 0.7167

## Model 2:

## Description:

The first layer is a **Flatten** layer, which takes an input with a shape of (100, 100, 1) and flattens it into a 1D array of size 10000. The input shape of (100, 100, 1) corresponds to 100x100 images with a single channel (e.g. grayscale).

The second layer is a **Dense** layer with 500 units and a ReLU activation function. The ReLU (Rectified Linear Unit) activation function is defined as f(x) = max(0, x). It is widely used in deep learning models because it helps improve the model's performance and training speed by eliminating the vanishing gradient problem.

The third layer is a **Dense** layer with 10 units and a softmax activation function. The softmax activation function maps the inputs to a probability distribution over the output classes. It is often used in the output layer of a classifier when there are multiple classes.

Calculating MSE and MAE of Model 2

MSE: 27.40

MAE: 4.60

```
Classification Report
              precision    recall  f1-score   support

           0       0.57      0.89      0.70         9
           1       0.86      0.75      0.80         8
           2       0.82      1.00      0.90         9
           3       0.88      0.78      0.82         9
           4       0.55      0.60      0.57        10
           5       0.58      0.88      0.70         8
           6       1.00      0.33      0.50         9
           7       0.80      0.50      0.62         8
           8       0.62      0.83      0.71        18
           9       0.80      0.33      0.47        12

    accuracy                           0.69       100
   macro avg       0.75      0.69      0.68       100
weighted avg       0.74      0.69      0.67       100
```

loss: 0.9858 - accuracy: 0.7191

## Model 3:

## Description:

The first layer is a **Flatten** layer, which takes an input with a shape of (100, 100, 1) and flattens it into a 1D array of size 10000. The input shape of (100, 100, 1) corresponds to 100x100 images with a single channel (e.g. grayscale).

The second layer is a **Dense** layer with 500 units and a ReLU activation function. The ReLU (Rectified Linear Unit) activation function is defined as f(x) = max(0, x). It is widely used in deep learning models because it helps improve the model's performance and training speed by eliminating the vanishing gradient problem.

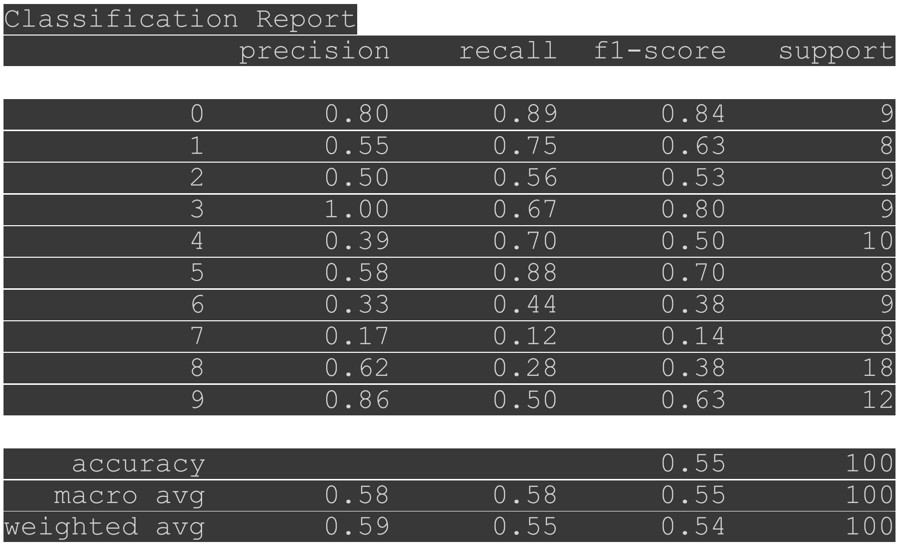
The third layer is a **Dense** layer with 1000 units and a ReLU activation function. This layer increases the model's capacity to learn and model more complex relationships in the data.

The fourth layer is a **Dense** layer with 10 units and a softmax activation function. The softmax activation function maps the inputs to a probability distribution over the output classes. It is often used in the output layer of a classifier when there are multiple classes.

Calculating MSE and MAE of Model 3

MSE: 32.35

MAE: 4.87



```
Classification Report
              precision    recall  f1-score   support

           0       0.80      0.89      0.84         9
           1       0.55      0.75      0.63         8
           2       0.50      0.56      0.53         9
           3       1.00      0.67      0.80         9
           4       0.39      0.70      0.50        10
           5       0.58      0.88      0.70         8
           6       0.33      0.44      0.38         9
           7       0.17      0.12      0.14         8
           8       0.62      0.28      0.38        18
           9       0.86      0.50      0.63        12

    accuracy                           0.55       100
   macro avg       0.58      0.58      0.55       100
weighted avg       0.59      0.55      0.54       100
```

loss: 1.0751 - accuracy: 0.5714

Conclusion:

Model 2 has the best accuracy

## CNN Model

Model 1 was NN not CNN but I was testing the accuracy.

## Model 2

## Description:

The model consists of eight layers:

A convolutional layer with 200 filters of size 3x3, using the ReLU activation function. This layer has an input shape of (100, 100, 3), which means it expects 100x100 pixel images with 3 channels (RGB).

A convolutional layer with 150 filters of size 3x3 and the ReLU activation function.

A max pooling layer with a pool size of 4x4. This layer reduces the size of the feature maps from the previous layer by taking the maximum value of each 4x4 block.

A convolutional layer with 120 filters of size 3x3 and the ReLU activation function.

A convolutional layer with 80 filters of size 3x3 and the ReLU activation function.

A convolutional layer with 50 filters of size 3x3 and the ReLU activation function.

Another max pooling layer with a pool size of 4x4.

A flatten layer that flattens the feature maps from the previous layer into a single one-dimensional vector.

A dense (fully connected) layer with 120 units and the ReLU activation function.

A dense layer with 100 units and the ReLU activation function.

A dense layer with 50 units and the ReLU activation function.

A dropout layer with a rate of 0.5, which randomly sets half of the input units to 0 at each update during training, in order to reduce overfitting.

A dense output layer with 6 units and the softmax activation function. This layer produces a probability distribution over the 6 classes.

Calculating MSE and MAE of Model 2

MSE: 26.83

MAE: 4.63

```
Classification Report
              precision    recall  f1-score   support

           0       1.00      0.89      0.94         9
           1       0.88      0.88      0.88         8
           2       0.90      1.00      0.95         9
           3       1.00      0.89      0.94         9
           4       0.67      0.80      0.73        10
           5       1.00      1.00      1.00         8
           6       0.78      0.78      0.78         9
           7       0.56      0.62      0.59         8
           8       1.00      0.83      0.91        18
           9       0.92      1.00      0.96        12

    accuracy                           1.00       100
   macro avg       0.87      0.87      1.00       100
```

Accuracy: 1.00

## Model 3

### Description

The third layer is another **Conv2D** layer with 50 filters and a kernel size of (3, 3). The ReLU activation function is applied to the output of this layer.

The fourth layer is another **MaxPool2D** layer with a pool size of (4, 4).

The fifth layer is a **Flatten** layer, which flattens the input into a 1D array.

The sixth layer is a **Dense** layer with 50 units and a ReLU activation function.

The seventh layer is a **Dropout** layer with a rate of 0.5. This layer randomly sets half of the input units to 0 at each update during training, which helps prevent overfitting.

The eighth and final layer is a **Dense** layer with 10 units and a softmax activation function. The softmax activation function maps the inputs to a probability distribution over the output classes. It is often used in the output layer of a classifier when there are multiple classes.

Calculating MSE and MAE of Model 3

MSE: 27.40

MAE: 4.60

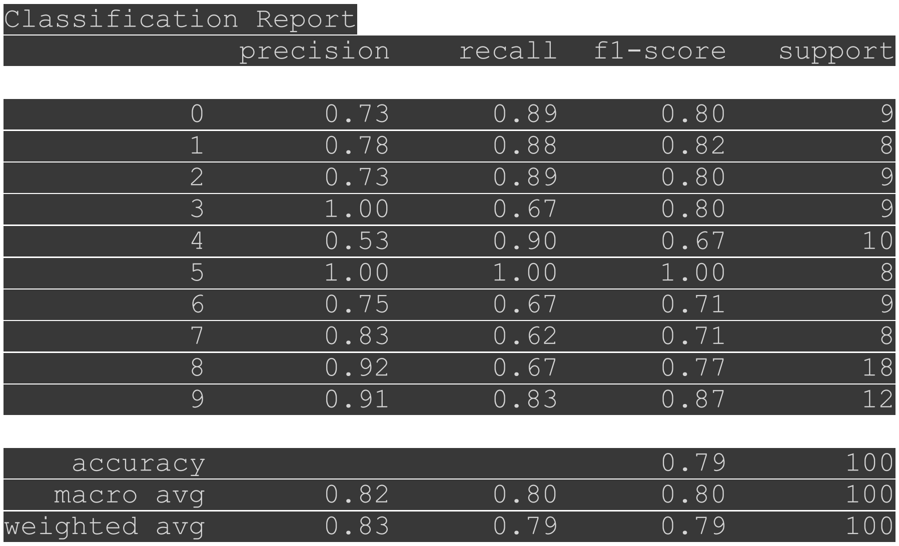
```
Classification Report
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         9
           1       0.89      1.00      0.94         8
           2       1.00      1.00      1.00         9
           3       1.00      0.89      0.94         9
           4       0.77      1.00      0.87        10
           5       1.00      1.00      1.00         8
           6       0.89      0.89      0.89         9
           7       1.00      0.88      0.93         8
           8       1.00      0.89      0.94        18
           9       1.00      1.00      1.00        12

    accuracy                           0.95       100
   macro avg       0.95      0.95      0.95       100
weighted avg       0.96      0.95      0.95       100
```

Accuracy: 0.95

## **SVM:**

### **Description:**

The model is then fit to the training data using the **fit** method. The **x_train** argument is the training input data, and the **y_train** argument is the corresponding training labels. It's worth noting that the input data has been reshaped to a 2D array with dimensions (1649, 3*100*100). This is likely done to ensure that the data has the correct shape for input into the SVM model.



```
Classification Report
              precision    recall  f1-score   support

           0       0.73      0.89      0.80         9
           1       0.78      0.88      0.82         8
           2       0.73      0.89      0.80         9
           3       1.00      0.67      0.80         9
           4       0.53      0.90      0.67        10
           5       1.00      1.00      1.00         8
           6       0.75      0.67      0.71         9
           7       0.83      0.62      0.71         8
           8       0.92      0.67      0.77        18
           9       0.91      0.83      0.87        12

    accuracy                           0.79       100
   macro avg       0.82      0.80      0.80       100
weighted avg       0.83      0.79      0.79       100
```

Accuracy on unknown data is 0.8184019370460048

- ❖ **We deduce that CNN has a better performance than SVM.**
- ❖ **All those model is done by using cross validation with 5 splits and 10 epochs.**