Importing the Libraries

```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib import pyplot
from numpy import unique, argmax
from tensorflow.keras.datasets.mnist import load_data
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Conv2D
from tensorflow.keras.layers import MaxPooling2D
from tensorflow.keras.layers import Flatten
from tensorflow.keras.layers import Dropout
from tensorflow.keras.utils import plot_model
```

Loading the MNIST Dataset

```
(x_train,y_train),(x_test,y_test)=load_data()
x_train=x_train.reshape((x_train.shape[0],x_train.shape[1],x_train.shape[2],1))
x_test=x_test.reshape((x_test.shape[0],x_test.shape[1],x_test.shape[2],1))
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz
11490434/11490434 ━━━━━━━━━━━━━━━━━━━━ 0s 0us/step
```

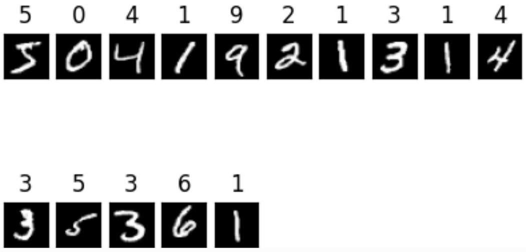
Normalizing the value of Pixels of images

```
x_train=x_train.astype('float32')/255
x_test=x_test.astype('float32')/255
```

Plotting the images

```
fig=plt.figure(figsize=(5,3))
fig=plt.figure(figsize=(5,3))
for i in range (15):
  ax=fig.add_subplot(2,10,i+1,xticks=[],yticks=[])
  ax.imshow(np.squeeze(x_train[i]),cmap='gray')
  ax.set_title(y_train[i])
```

```
<Figure size 500x300 with 0 Axes>
```



Determining the shape of Input images

```
img_shape=x_train.shape[1:]
print(img_shape)
```

```
(28, 28, 1)
```

Defining the Model (2D Convolutional Layer)

```
model=Sequential()
model.add(Conv2D(32,(3,3),activation='relu',input_shape=img_shape))
model.add(MaxPooling2D((2,2)))
model.add(Conv2D(48,(3,3),activation='relu'))
model.add(Dropout((0.5)))
model.add(Flatten())
model.add(Dense(500,activation='relu'))
model.add(Dense(10,activation='softmax'))
```

```
/usr/local/lib/python3.11/dist-packages/keras/src/layers/convolutional/base_conv.py:107: UserWarning: Do not pass an `input_shape`/`inpu
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

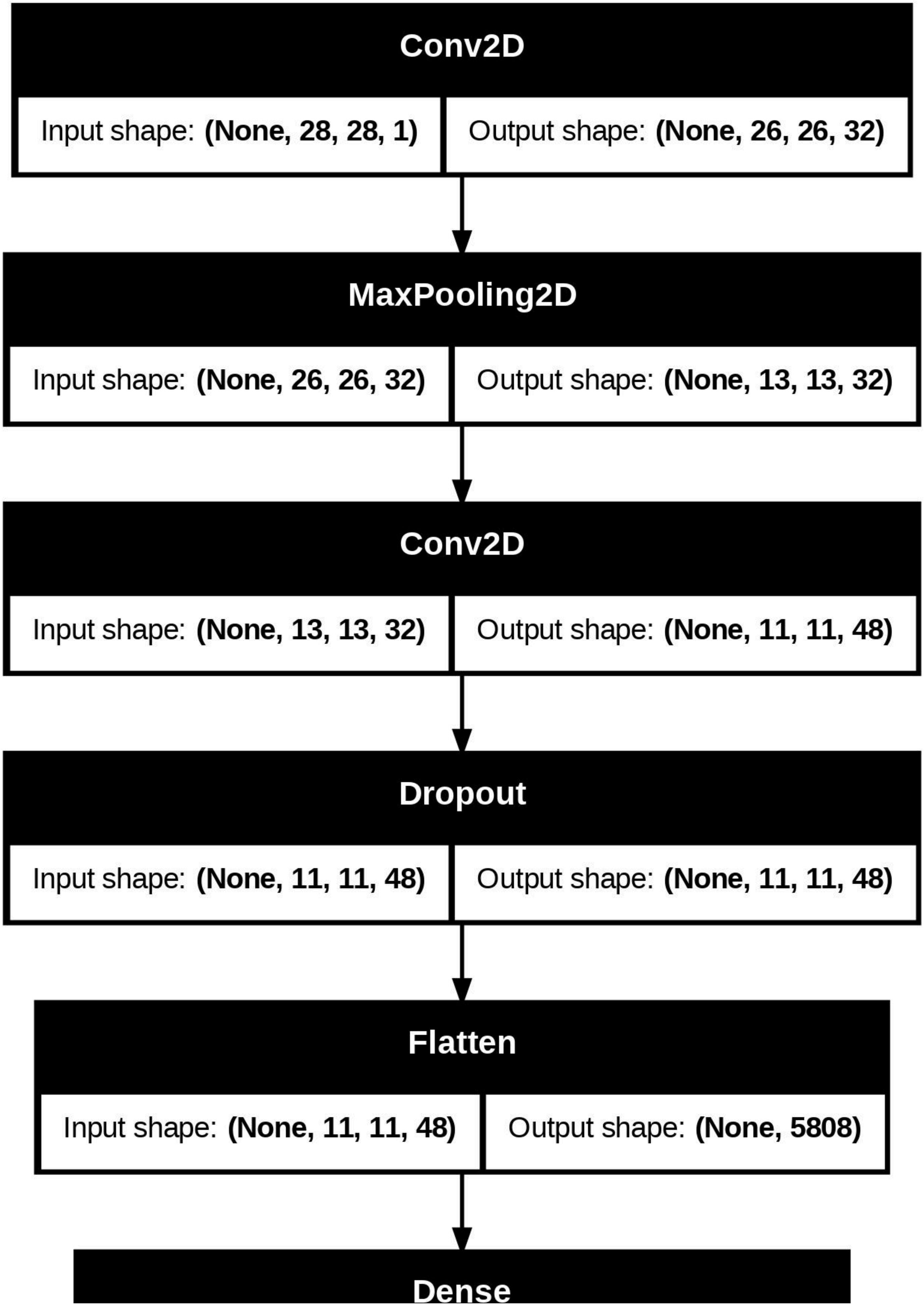
Model Summary

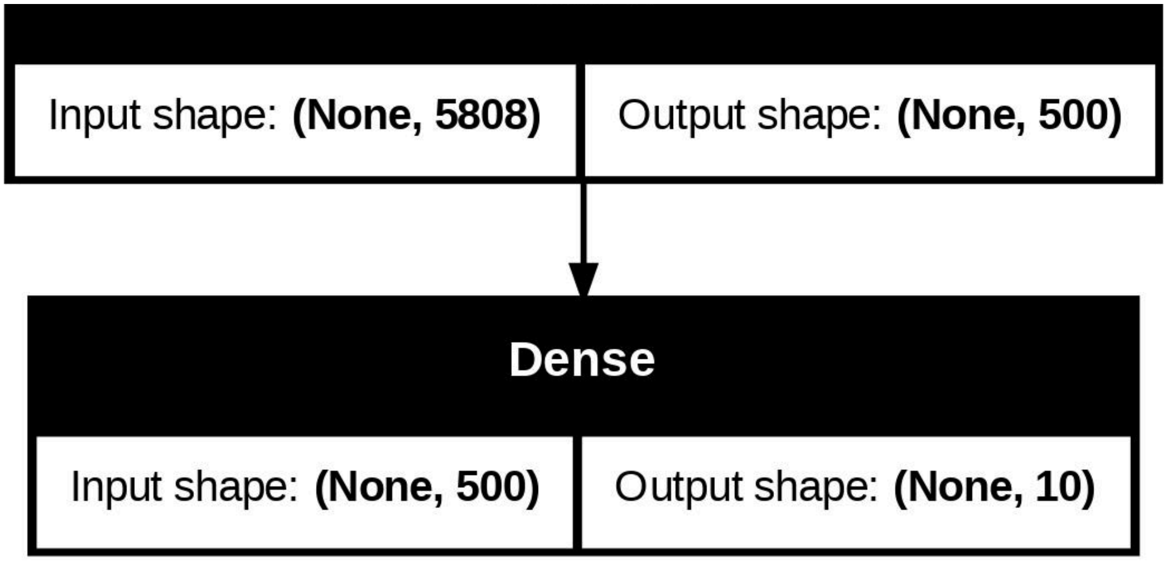
```
model.summary()
```

```
Model: "sequential"
┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━┓
┃ Layer (type)                         ┃ Output Shape                ┃         Param # ┃
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━┩
│ conv2d (Conv2D)                      │ (None, 26, 26, 32)          │             320 │
├──────────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ max_pooling2d (MaxPooling2D)         │ (None, 13, 13, 32)          │               0 │
├──────────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ conv2d_1 (Conv2D)                    │ (None, 11, 11, 48)          │          13,872 │
├──────────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ dropout (Dropout)                    │ (None, 11, 11, 48)          │               0 │
├──────────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ flatten (Flatten)                    │ (None, 5808)                │               0 │
├──────────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ dense (Dense)                        │ (None, 500)                 │       2,904,500 │
├──────────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ dense_1 (Dense)                      │ (None, 10)                  │           5,010 │
└──────────────────────────────────────┴─────────────────────────────┴─────────────────┘
 Total params: 2,923,702 (11.15 MB)
 Trainable params: 2,923,702 (11.15 MB)
 Non-trainable params: 0 (0.00 B)
```

Showing the model shapes

```
plot_model(model,'model.jpg',show_shapes=True)
```


Conv2D
Input shape: (None, 28, 28, 1)
Output shape: (None, 26, 26, 32)
MaxPooling2D
Input shape: (None, 26, 26, 32)
Output shape: (None, 13, 13, 32)
Conv2D
Input shape: (None, 13, 13, 32)
Output shape: (None, 11, 11, 48)
Dropout
Input shape: (None, 11, 11, 48)
Output shape: (None, 11, 11, 48)
Flatten
Input shape: (None, 11, 11, 48)
Output shape: (None, 5808)
Dense


Input shape: (None, 5808)
Output shape: (None, 500)
Dense
Input shape: (None, 500)
Output shape: (None, 10)

Training and validation of Model

```
model.compile(optimizer='adam',loss='sparse_categorical_crossentropy',metrics=['accuracy'])
x=model.fit(x_train,y_train,epochs=10,batch_size=128,verbose=2,validation_split=0.1)
```

```
Epoch 1/10
422/422 - 66s - 157ms/step - accuracy: 0.9476 - loss: 0.1722 - val_accuracy: 0.9858 - val_loss: 0.0502
Epoch 2/10
422/422 - 84s - 200ms/step - accuracy: 0.9831 - loss: 0.0554 - val_accuracy: 0.9892 - val_loss: 0.0390
Epoch 3/10
422/422 - 65s - 154ms/step - accuracy: 0.9882 - loss: 0.0390 - val_accuracy: 0.9900 - val_loss: 0.0364
Epoch 4/10
422/422 - 82s - 195ms/step - accuracy: 0.9907 - loss: 0.0286 - val_accuracy: 0.9930 - val_loss: 0.0288
Epoch 5/10
422/422 - 66s - 157ms/step - accuracy: 0.9924 - loss: 0.0227 - val_accuracy: 0.9908 - val_loss: 0.0362
Epoch 6/10
422/422 - 86s - 205ms/step - accuracy: 0.9938 - loss: 0.0181 - val_accuracy: 0.9917 - val_loss: 0.0353
Epoch 7/10
```