

In [2]:

```
from keras.layers import *
from keras.models import Sequential
```

Building Model

In [3]:

```
model = Sequential()
model.add(Conv2D(32,(3,3),activation='relu',input_shape=(28,28,1)))
model.add(MaxPool2D((2,2)))
model.add(Conv2D(64,(3,3),activation='relu'))
model.add(MaxPool2D((2,2)))
model.add(Conv2D(64,(3,3),activation='relu'))
model.add(Flatten())
model.add(Dense(64,activation='relu'))
model.add(Dense(10,activation='softmax'))
```

In [4]:

```
model.summary()
```

Model: "sequential_1"

Layer (type)	Output Shape	Param #
=====		
conv2d_1 (Conv2D)	(None, 26, 26, 32)	320

max_pooling2d_1 (MaxPooling2	(None, 13, 13, 32)	0

conv2d_2 (Conv2D)	(None, 11, 11, 64)	18496

max_pooling2d_2 (MaxPooling2	(None, 5, 5, 64)	0

conv2d_3 (Conv2D)	(None, 3, 3, 64)	36928

flatten_1 (Flatten)	(None, 576)	0

dense_1 (Dense)	(None, 64)	36928

dense_2 (Dense)	(None, 10)	650
=====		
Total params: 93,322		
Trainable params: 93,322		
Non-trainable params: 0		

Importing Dataset

In [5]:

```
from keras.datasets import mnist
from keras.utils import to_categorical
```

In [6]:

```
(X_Train,Y_Train),(X_Test,Y_Test) = mnist.load_data()
```

In [7]:

```
def preprocess_data(X,Y):
    X = X.reshape((-1,28,28,1))
    X = X/255.0
    Y = to_categorical(Y)
    return X,Y
```

In [8]:

```
X_Train, Y_Train = preprocess_data(X_Train,Y_Train)
print(X_Train.shape,Y_Train.shape)
```

```
(60000, 28, 28, 1) (60000, 10)
```

In [9]:

```
X_Test, Y_Test = preprocess_data(X_Test,Y_Test)
print(X_Test.shape,Y_Test.shape)
```

```
(10000, 28, 28, 1) (10000, 10)
```

Compiling Model

In [10]:

```
model.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accuracy'])
```

In [11]:

```
hist = model.fit(X_Train,Y_Train,epochs=20,validation_split=0.1,batch_size=128)
```

Train on 54000 samples, validate on 6000 samples

Epoch 1/20

54000/54000 [=====] - 39s 719us/step - loss: 0.2739 - accuracy: 0.9214 - val_loss: 0.0

Epoch 2/20

54000/54000 [=====] - 35s 649us/step - loss: 0.0662 - accuracy: 0.9793 - val_loss: 0.0

Epoch 3/20

54000/54000 [=====] - 34s 637us/step - loss: 0.0461 - accuracy: 0.9856 - val_loss: 0.0

Epoch 4/20

54000/54000 [=====] - 36s 666us/step - loss: 0.0359 - accuracy: 0.9885 - val_loss: 0.0

Epoch 5/20

54000/54000 [=====] - 40s 735us/step - loss: 0.0290 - accuracy: 0.9911 - val_loss: 0.0

Epoch 6/20

54000/54000 [=====] - 39s 724us/step - loss: 0.0230 - accuracy: 0.9928 - val_loss: 0.0

Epoch 7/20

54000/54000 [=====] - 40s 748us/step - loss: 0.0203 - accuracy: 0.9938 - val_loss: 0.0

Epoch 8/20

54000/54000 [=====] - 43s 797us/step - loss: 0.0170 - accuracy: 0.9944 - val_loss: 0.0

Epoch 9/20

54000/54000 [=====] - 42s 784us/step - loss: 0.0133 - accuracy: 0.9956 - val_loss: 0.0

Epoch 10/20

54000/54000 [=====] - 38s 710us/step - loss: 0.0135 - accuracy: 0.9956 - val_loss: 0.0

Epoch 11/20

54000/54000 [=====] - 40s 742us/step - loss: 0.0111 - accuracy: 0.9963 - val_loss: 0.0

Epoch 12/20

54000/54000 [=====] - 34s 638us/step - loss: 0.0108 - accuracy: 0.9965 - val_loss: 0.0

Epoch 13/20

54000/54000 [=====] - 36s 657us/step - loss: 0.0068 - accuracy: 0.9977 - val_loss: 0.0

Epoch 14/20

54000/54000 [=====] - 37s 692us/step - loss: 0.0070 - accuracy: 0.9975 - val_loss: 0.0

Epoch 15/20

54000/54000 [=====] - 39s 729us/step - loss: 0.0079 - accuracy: 0.9974 - val_loss: 0.0

Epoch 16/20

54000/54000 [=====] - 37s 677us/step - loss: 0.0061 - accuracy: 0.9979 - val_loss: 0.0

Epoch 17/20

54000/54000 [=====] - 40s 734us/step - loss: 0.0062 - accuracy: 0.9978 - val_loss: 0.0

Epoch 18/20

54000/54000 [=====] - 36s 668us/step - loss: 0.0071 - accuracy: 0.9979 - val_loss: 0.0

Epoch 19/20

54000/54000 [=====] - 35s 644us/step - loss: 0.0040 - accuracy: 0.9988 - val_loss: 0.0

Epoch 20/20

54000/54000 [=====] - 37s 677us/step - loss: 0.0044 - accuracy: 0.9986 - val_loss: 0.0

In [12]:

```
model.evaluate(X_Test,Y_Test)
```

10000/10000 [=====] - 2s 201us/step

[0.03887164531196565, 0.9902999997138977]

```
In [ ]:
```