Importing necessary Modules

In [ ]:
```
import tensorflow as tf
from tensorflow.keras import Input, Model
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPool2D, Dense, Flatten, Conv2DTr
from tensorflow.keras.datasets import cifar10
from tensorflow.keras.losses import mean_squared_error
from tensorflow.keras.layers import BatchNormalization
import matplotlib.pyplot as plt
from tensorflow.keras.layers import Reshape
import numpy as np
```

In [ ]:
```
(x_train, _), (x_test, _) = cifar10.load_data()
print(x_train.shape)
print(x_test.shape)

x_train = x_train.astype('float32') / 255.0
print("The training data is",x_train)
x_test = x_test.astype('float32') / 255.0
print("The testing data is",x_test)

# Split the data into training, validation, and test sets

validation_split = 0.1
validation_size = int(validation_split * len(x_train))
print ("The Validation data size is",validation_size)
train_data = x_train[:-validation_size]
val_data = x_train[-validation_size:]
test_data = x_test
```

```
(50000, 32, 32, 3)
(10000, 32, 32, 3)
The training data is [[[[0.23137255 0.24313726 0.24705882]
   [0.16862746 0.18039216 0.1764706 ]
   [0.19607843 0.1882353  0.16862746]
   ...
   [0.61960787 0.5176471  0.42352942]
   [0.59607846 0.49019608 0.4       ]
   [0.5803922  0.4862745  0.40392157]]

  [[0.0627451  0.07843138 0.07843138]
   [0.         0.         0.        ]
   [0.07058824 0.03137255 0.        ]
   ...
   [0.48235294 0.34509805 0.21568628]
   [0.46666667 0.3254902  0.19607843]
   [0.47843137 0.34117648 0.22352941]]

  [[0.09803922 0.09411765 0.08235294]
   [0.0627451  0.02745098 0.        ]
   [0.19215687 0.10588235 0.03137255]
   ...
   [0.4627451  0.32941177 0.19607843]
   [0.47058824 0.32941177 0.19607843]
   [0.42745098 0.28627452 0.16470589]]

  ...

  [[0.8156863  0.6666667  0.3764706 ]
   [0.7882353  0.6        0.13333334]
   [0.7764706  0.6313726  0.10196079]
   ...
   [0.627451   0.52156866 0.27450982]
   [0.21960784 0.12156863 0.02745098]
   [0.20784314 0.13333334 0.07843138]]

  [[0.7058824  0.54509807 0.3764706 ]
   [0.6784314  0.48235294 0.16470589]
   [0.7294118  0.5647059  0.11764706]
   ...
   [0.72156864 0.5803922  0.36862746]
   [0.38039216 0.24313726 0.13333334]
   [0.3254902  0.20784314 0.13333334]]

  [[0.69411767 0.5647059  0.45490196]
   [0.65882355 0.5058824  0.36862746]
   [0.7019608  0.5568628  0.34117648]
   ...
   [0.84705883 0.72156864 0.54901963]
   [0.5921569  0.4627451  0.32941177]
   [0.48235294 0.36078432 0.28235295]]]


 [[[0.6039216  0.69411767 0.73333335]
   [0.49411765 0.5372549  0.53333336]
   [0.4117647  0.40784314 0.37254903]
   ...
   [0.35686275 0.37254903 0.2784314 ]
   [0.34117648 0.3529412  0.2784314 ]
   [0.30980393 0.31764707 0.27450982]]
```

```
  [[0.54901963 0.627451   0.6627451 ]
   [0.5686275  0.6        0.6039216 ]
   [0.49019608 0.49019608 0.4627451 ]
   ...
   [0.3764706  0.3882353  0.30588236]
   [0.3019608  0.3137255  0.24313726]
   [0.2784314  0.28627452 0.23921569]]

  [[0.54901963 0.60784316 0.6431373 ]
   [0.54509807 0.57254905 0.58431375]
   [0.4509804  0.4509804  0.4392157 ]
   ...
   [0.30980393 0.32156864 0.2509804 ]
   [0.26666668 0.27450982 0.21568628]
   [0.2627451  0.27058825 0.21568628]]

  ...

  [[0.6862745  0.654902   0.6509804 ]
   [0.6117647  0.6039216  0.627451  ]
   [0.6039216  0.627451   0.6666667 ]
   ...
   [0.16470589 0.13333334 0.14117648]
   [0.23921569 0.20784314 0.22352941]
   [0.3647059  0.3254902  0.35686275]]

  [[0.64705884 0.6039216  0.5019608 ]
   [0.6117647  0.59607846 0.50980395]
   [0.62352943 0.6313726  0.5568628 ]
   ...
   [0.40392157 0.3647059  0.3764706 ]
   [0.48235294 0.44705883 0.47058824]
   [0.5137255  0.4745098  0.5137255 ]]

  [[0.6392157  0.5803922  0.47058824]
   [0.61960787 0.5803922  0.47843137]
   [0.6392157  0.6117647  0.52156866]
   ...
   [0.56078434 0.52156866 0.54509807]
   [0.56078434 0.5254902  0.5568628 ]
   [0.56078434 0.52156866 0.5647059 ]]]


 [[[1.         1.         1.        ]
   [0.99215686 0.99215686 0.99215686]
   [0.99215686 0.99215686 0.99215686]
   ...
   [0.99215686 0.99215686 0.99215686]
   [0.99215686 0.99215686 0.99215686]
   [0.99215686 0.99215686 0.99215686]]

  [[1.         1.         1.        ]
   [1.         1.         1.        ]
   [1.         1.         1.        ]
   ...
   [1.         1.         1.        ]
   [1.         1.         1.        ]
   [1.         1.         1.        ]]
```

```
  [[1.         1.         1.        ]
   [0.99607843 0.99607843 0.99607843]
   [0.99607843 0.99607843 0.99607843]
   ...
   [0.99607843 0.99607843 0.99607843]
   [0.99607843 0.99607843 0.99607843]
   [0.99607843 0.99607843 0.99607843]]

  ...

  [[0.44313726 0.47058824 0.4392157 ]
   [0.43529412 0.4627451  0.43529412]
   [0.4117647  0.4392157  0.41568628]
   ...
   [0.28235295 0.31764707 0.3137255 ]
   [0.28235295 0.3137255  0.30980393]
   [0.28235295 0.3137255  0.30980393]]

  [[0.43529412 0.4627451  0.43137255]
   [0.40784314 0.43529412 0.40784314]
   [0.3882353  0.41568628 0.38431373]
   ...
   [0.26666668 0.29411766 0.28627452]
   [0.27450982 0.29803923 0.29411766]
   [0.30588236 0.32941177 0.32156864]]

  [[0.41568628 0.44313726 0.4117647 ]
   [0.3882353  0.41568628 0.38431373]
   [0.37254903 0.4        0.36862746]
   ...
   [0.30588236 0.33333334 0.3254902 ]
   [0.30980393 0.33333334 0.3254902 ]
   [0.3137255  0.3372549  0.32941177]]]


 ...


 [[[0.13725491 0.69803923 0.92156863]
   [0.15686275 0.6901961  0.9372549 ]
   [0.16470589 0.6901961  0.94509804]
   ...
   [0.3882353  0.69411767 0.85882354]
   [0.30980393 0.5764706  0.77254903]
   [0.34901962 0.5803922  0.7411765 ]]

  [[0.22352941 0.7137255  0.91764706]
   [0.17254902 0.72156864 0.98039216]
   [0.19607843 0.7176471  0.9411765 ]
   ...
   [0.6117647  0.7137255  0.78431374]
   [0.5529412  0.69411767 0.80784315]
   [0.45490196 0.58431375 0.6862745 ]]

  [[0.38431373 0.77254903 0.92941177]
   [0.2509804  0.7411765  0.9882353 ]
   [0.27058825 0.7529412  0.9607843 ]
   ...
   [0.7372549  0.7647059  0.80784315]
   [0.46666667 0.5294118  0.5764706 ]
```

```
  [0.23921569 0.30980393 0.3529412 ]]

 ...

 [[0.28627452 0.30980393 0.3019608 ]
  [0.20784314 0.24705882 0.26666668]
  [0.21176471 0.26666668 0.3137255 ]
  ...
  [0.06666667 0.15686275 0.2509804 ]
  [0.08235294 0.14117648 0.2       ]
  [0.12941177 0.1882353  0.19215687]]

 [[0.23921569 0.26666668 0.29411766]
  [0.21568628 0.27450982 0.3372549 ]
  [0.22352941 0.30980393 0.40392157]
  ...
  [0.09411765 0.1882353  0.28235295]
  [0.06666667 0.13725491 0.20784314]
  [0.02745098 0.09019608 0.1254902 ]]

 [[0.17254902 0.21960784 0.28627452]
  [0.18039216 0.25882354 0.34509805]
  [0.19215687 0.3019608  0.4117647 ]
  ...
  [0.10588235 0.20392157 0.3019608 ]
  [0.08235294 0.16862746 0.25882354]
  [0.04705882 0.12156863 0.19607843]]]


[[[0.7411765  0.827451   0.9411765 ]
  [0.7294118  0.8156863  0.9254902 ]
  [0.7254902  0.8117647  0.92156863]
  ...
  [0.6862745  0.7647059  0.8784314 ]
  [0.6745098  0.7607843  0.87058824]
  [0.6627451  0.7607843  0.8627451 ]]

 [[0.7607843  0.8235294  0.9372549 ]
  [0.7490196  0.8117647  0.9254902 ]
  [0.74509805 0.80784315 0.92156863]
  ...
  [0.6784314  0.7529412  0.8627451 ]
  [0.67058825 0.7490196  0.85490197]
  [0.654902   0.74509805 0.84705883]]

 [[0.8156863  0.85882354 0.95686275]
  [0.8039216  0.84705883 0.9411765 ]
  [0.8        0.84313726 0.9372549 ]
  ...
  [0.6862745  0.7490196  0.8509804 ]
  [0.6745098  0.74509805 0.84705883]
  [0.6627451  0.7490196  0.84313726]]

 ...

 [[0.8117647  0.78039217 0.70980394]
  [0.79607844 0.7647059  0.6862745 ]
  [0.79607844 0.76862746 0.6784314 ]
  ...
  [0.5294118  0.5176471  0.49803922]
```

```
   [0.63529414 0.61960787 0.5882353 ]
   [0.65882355 0.6392157  0.5921569 ]]

  [[0.7764706  0.74509805 0.6666667 ]
   [0.7411765  0.70980394 0.62352943]
   [0.7058824  0.6745098  0.5764706 ]
   ...
   [0.69803923 0.67058825 0.627451  ]
   [0.6862745  0.6627451  0.6117647 ]
   [0.6862745  0.6627451  0.6039216 ]]

  [[0.7764706  0.7411765  0.6784314 ]
   [0.7411765  0.70980394 0.63529414]
   [0.69803923 0.6666667  0.58431375]
   ...
   [0.7647059  0.72156864 0.6627451 ]
   [0.76862746 0.7411765  0.67058825]
   [0.7647059  0.74509805 0.67058825]]]


 [[[0.8980392  0.8980392  0.9372549 ]
   [0.9254902  0.92941177 0.96862745]
   [0.91764706 0.9254902  0.96862745]
   ...
   [0.8509804  0.85882354 0.9137255 ]
   [0.8666667  0.8745098  0.91764706]
   [0.87058824 0.8745098  0.9137255 ]]

  [[0.87058824 0.8666667  0.8980392 ]
   [0.9372549  0.9372549  0.9764706 ]
   [0.9137255  0.91764706 0.9647059 ]
   ...
   [0.8745098  0.8745098  0.9254902 ]
   [0.8901961  0.89411765 0.93333334]
   [0.8235294  0.827451   0.8627451 ]]

  [[0.8352941  0.80784315 0.827451  ]
   [0.91764706 0.9098039  0.9372549 ]
   [0.90588236 0.9137255  0.95686275]
   ...
   [0.8627451  0.8627451  0.9098039 ]
   [0.8627451  0.85882354 0.9098039 ]
   [0.7921569  0.79607844 0.84313726]]

  ...

  [[0.5882353  0.56078434 0.5294118 ]
   [0.54901963 0.5294118  0.49803922]
   [0.5176471  0.49803922 0.47058824]
   ...
   [0.8784314  0.87058824 0.85490197]
   [0.9019608  0.89411765 0.88235295]
   [0.94509804 0.94509804 0.93333334]]

  [[0.5372549  0.5176471  0.49411765]
   [0.50980395 0.49803922 0.47058824]
   [0.49019608 0.4745098  0.4509804 ]
   ...
   [0.70980394 0.7058824  0.69803923]
   [0.7921569  0.7882353  0.7764706 ]
```

```
   [0.83137256 0.827451   0.8117647 ]]

  [[0.47843137 0.46666667 0.44705883]
   [0.4627451  0.45490196 0.43137255]
   [0.47058824 0.45490196 0.43529412]
   ...
   [0.7019608  0.69411767 0.6784314 ]
   [0.6431373  0.6431373  0.63529414]
   [0.6392157  0.6392157  0.6313726 ]]]]
The testing data is [[[[0.61960787 0.4392157  0.19215687]
   [0.62352943 0.43529412 0.18431373]
   [0.64705884 0.45490196 0.2       ]
   ...
   [0.5372549  0.37254903 0.14117648]
   [0.49411765 0.35686275 0.14117648]
   [0.45490196 0.33333334 0.12941177]]

  [[0.59607846 0.4392157  0.2       ]
   [0.5921569  0.43137255 0.15686275]
   [0.62352943 0.44705883 0.1764706 ]
   ...
   [0.53333336 0.37254903 0.12156863]
   [0.49019608 0.35686275 0.1254902 ]
   [0.46666667 0.34509805 0.13333334]]

  [[0.5921569  0.43137255 0.18431373]
   [0.5921569  0.42745098 0.12941177]
   [0.61960787 0.43529412 0.14117648]
   ...
   [0.54509807 0.38431373 0.13333334]
   [0.50980395 0.37254903 0.13333334]
   [0.47058824 0.34901962 0.12941177]]

  ...

  [[0.26666668 0.4862745  0.69411767]
   [0.16470589 0.39215687 0.5803922 ]
   [0.12156863 0.34509805 0.5372549 ]
   ...
   [0.14901961 0.38039216 0.57254905]
   [0.05098039 0.2509804  0.42352942]
   [0.15686275 0.33333334 0.49803922]]

  [[0.23921569 0.45490196 0.65882355]
   [0.19215687 0.4        0.5803922 ]
   [0.13725491 0.33333334 0.5176471 ]
   ...
   [0.10196079 0.32156864 0.50980395]
   [0.11372549 0.32156864 0.49411765]
   [0.07843138 0.2509804  0.41960785]]

  [[0.21176471 0.41960785 0.627451  ]
   [0.21960784 0.4117647  0.58431375]
   [0.1764706  0.34901962 0.5176471 ]
   ...
   [0.09411765 0.3019608  0.4862745 ]
   [0.13333334 0.32941177 0.5058824 ]
   [0.08235294 0.2627451  0.43137255]]]
```

```
[[[0.92156863 0.92156863 0.92156863]
  [0.90588236 0.90588236 0.90588236]
  [0.9098039  0.9098039  0.9098039 ]
  ...
  [0.9137255  0.9137255  0.9137255 ]
  [0.9137255  0.9137255  0.9137255 ]
  [0.9098039  0.9098039  0.9098039 ]]

 [[0.93333334 0.93333334 0.93333334]
  [0.92156863 0.92156863 0.92156863]
  [0.92156863 0.92156863 0.92156863]
  ...
  [0.9254902  0.9254902  0.9254902 ]
  [0.9254902  0.9254902  0.9254902 ]
  [0.92156863 0.92156863 0.92156863]]

 [[0.92941177 0.92941177 0.92941177]
  [0.91764706 0.91764706 0.91764706]
  [0.91764706 0.91764706 0.91764706]
  ...
  [0.92156863 0.92156863 0.92156863]
  [0.92156863 0.92156863 0.92156863]
  [0.91764706 0.91764706 0.91764706]]

 ...

 [[0.34117648 0.3882353  0.34901962]
  [0.16862746 0.2        0.14509805]
  [0.07450981 0.09019608 0.04313726]
  ...
  [0.6627451  0.72156864 0.7019608 ]
  [0.7137255  0.77254903 0.75686276]
  [0.7372549  0.7921569  0.7882353 ]]

 [[0.32156864 0.3764706  0.32156864]
  [0.18039216 0.22352941 0.14117648]
  [0.14117648 0.17254902 0.08627451]
  ...
  [0.68235296 0.7411765  0.7176471 ]
  [0.7254902  0.78431374 0.76862746]
  [0.73333335 0.7921569  0.78431374]]

 [[0.33333334 0.39607844 0.3254902 ]
  [0.24313726 0.29411766 0.1882353 ]
  [0.22745098 0.2627451  0.14901961]
  ...
  [0.65882355 0.7176471  0.69803923]
  [0.7058824  0.7647059  0.7490196 ]
  [0.7294118  0.78431374 0.78039217]]]


[[[0.61960787 0.74509805 0.87058824]
  [0.61960787 0.73333335 0.85490197]
  [0.54509807 0.6509804  0.7607843 ]
  ...
  [0.89411765 0.90588236 0.91764706]
  [0.92941177 0.9372549  0.9529412 ]
  [0.93333334 0.94509804 0.9647059 ]]

 [[0.6666667  0.78431374 0.8980392 ]
```

```
  [0.6745098  0.78039217 0.8862745 ]
  [0.5921569  0.6901961  0.7882353 ]
  ...
  [0.9098039  0.9098039  0.9254902 ]
  [0.9647059  0.9647059  0.98039216]
  [0.9647059  0.96862745 0.9843137 ]]

 [[0.68235296 0.7882353  0.88235295]
  [0.6901961  0.78431374 0.87058824]
  [0.6156863  0.7019608  0.78039217]
  ...
  [0.9019608  0.8980392  0.9098039 ]
  [0.98039216 0.9764706  0.9843137 ]
  [0.9607843  0.95686275 0.96862745]]

 ...

 [[0.12156863 0.15686275 0.1764706 ]
  [0.11764706 0.15294118 0.17254902]
  [0.10196079 0.13725491 0.15686275]
  ...
  [0.14509805 0.15686275 0.18039216]
  [0.03529412 0.05098039 0.05490196]
  [0.01568628 0.02745098 0.01960784]]

 [[0.09019608 0.13333334 0.15294118]
  [0.10588235 0.14901961 0.16862746]
  [0.09803922 0.14117648 0.16078432]
  ...
  [0.07450981 0.07843138 0.09411765]
  [0.01568628 0.02352941 0.01176471]
  [0.01960784 0.02745098 0.01176471]]

 [[0.10980392 0.16078432 0.18431373]
  [0.11764706 0.16862746 0.19607843]
  [0.1254902  0.1764706  0.20392157]
  ...
  [0.01960784 0.02352941 0.03137255]
  [0.01568628 0.01960784 0.01176471]
  [0.02745098 0.03137255 0.02745098]]]


...


[[[0.07843138 0.05882353 0.04705882]
  [0.07450981 0.05490196 0.04313726]
  [0.05882353 0.05490196 0.04313726]
  ...
  [0.03921569 0.03529412 0.02745098]
  [0.04705882 0.04313726 0.03529412]
  [0.05098039 0.04705882 0.03921569]]

 [[0.08235294 0.0627451  0.05098039]
  [0.07843138 0.0627451  0.05098039]
  [0.07058824 0.06666667 0.04705882]
  ...
  [0.03921569 0.03529412 0.02745098]
  [0.03921569 0.03529412 0.02745098]
  [0.04705882 0.04313726 0.03529412]]
```

```
  [[0.08235294 0.0627451  0.05098039]
   [0.08235294 0.06666667 0.04705882]
   [0.07843138 0.07058824 0.04313726]
   ...
   [0.04705882 0.04313726 0.03529412]
   [0.04705882 0.04313726 0.03529412]
   [0.05098039 0.04705882 0.03921569]]

  ...

  [[0.12941177 0.09803922 0.05098039]
   [0.13333334 0.10196079 0.05882353]
   [0.13333334 0.10196079 0.05882353]
   ...
   [0.10980392 0.09803922 0.20392157]
   [0.11372549 0.09803922 0.22745098]
   [0.09019608 0.07843138 0.16470589]]

  [[0.12941177 0.09803922 0.05490196]
   [0.13333334 0.10196079 0.05882353]
   [0.13333334 0.10196079 0.05882353]
   ...
   [0.10588235 0.09411765 0.20392157]
   [0.10588235 0.09411765 0.21960784]
   [0.09803922 0.08627451 0.18431373]]

  [[0.12156863 0.09019608 0.04705882]
   [0.1254902  0.09411765 0.05098039]
   [0.12941177 0.09803922 0.05490196]
   ...
   [0.09411765 0.09019608 0.19607843]
   [0.10196079 0.09019608 0.20784314]
   [0.09803922 0.07843138 0.18431373]]]


 [[[0.09803922 0.15686275 0.04705882]
   [0.05882353 0.14117648 0.01176471]
   [0.09019608 0.16078432 0.07058824]
   ...
   [0.23921569 0.32156864 0.30588236]
   [0.36078432 0.44313726 0.4392157 ]
   [0.29411766 0.34901962 0.36078432]]

  [[0.04705882 0.09803922 0.02352941]
   [0.07843138 0.14509805 0.02745098]
   [0.09411765 0.14117648 0.05882353]
   ...
   [0.4509804  0.5254902  0.5411765 ]
   [0.58431375 0.65882355 0.69411767]
   [0.40784314 0.45882353 0.5137255 ]]

  [[0.04705882 0.09803922 0.04313726]
   [0.05882353 0.11372549 0.02352941]
   [0.13333334 0.15686275 0.09411765]
   ...
   [0.6039216  0.6745098  0.7137255 ]
   [0.6156863  0.6862745  0.7529412 ]
   [0.45490196 0.5058824  0.5921569 ]]
```

```
  ...

  [[0.39215687 0.5058824  0.31764707]
   [0.40392157 0.5176471  0.32941177]
   [0.40784314 0.5254902  0.3372549 ]
   ...
   [0.38039216 0.5019608  0.32941177]
   [0.38431373 0.49411765 0.32941177]
   [0.35686275 0.4745098  0.30980393]]

  [[0.40392157 0.5176471  0.3254902 ]
   [0.40784314 0.5137255  0.3254902 ]
   [0.41960785 0.5294118  0.34117648]
   ...
   [0.39607844 0.5176471  0.34117648]
   [0.3882353  0.49803922 0.32941177]
   [0.36078432 0.4745098  0.30980393]]

  [[0.37254903 0.49411765 0.30588236]
   [0.37254903 0.48235294 0.29803923]
   [0.39607844 0.5019608  0.31764707]
   ...
   [0.3647059  0.4862745  0.3137255 ]
   [0.37254903 0.48235294 0.31764707]
   [0.36078432 0.47058824 0.3137255 ]]]


 [[[0.28627452 0.30588236 0.29411766]
   [0.38431373 0.40392157 0.44313726]
   [0.3882353  0.41568628 0.44705883]
   ...
   [0.5294118  0.5882353  0.59607846]
   [0.5294118  0.58431375 0.6039216 ]
   [0.79607844 0.84313726 0.8745098 ]]

  [[0.27058825 0.28627452 0.27450982]
   [0.32941177 0.34901962 0.38039216]
   [0.26666668 0.29411766 0.31764707]
   ...
   [0.33333334 0.37254903 0.34901962]
   [0.2784314  0.32156864 0.3137255 ]
   [0.47058824 0.52156866 0.5294118 ]]

  [[0.27058825 0.28627452 0.27450982]
   [0.3529412  0.37254903 0.39215687]
   [0.24313726 0.2784314  0.2901961 ]
   ...
   [0.2901961  0.31764707 0.27450982]
   [0.20784314 0.24313726 0.21176471]
   [0.24313726 0.2901961  0.27058825]]

  ...

  [[0.48235294 0.5019608  0.3764706 ]
   [0.5176471  0.5176471  0.4       ]
   [0.5058824  0.5019608  0.39215687]
   ...
   [0.42352942 0.41960785 0.34509805]
   [0.24313726 0.23529412 0.21568628]
   [0.10588235 0.10588235 0.10980392]]
```

```
  [[0.4509804  0.4745098  0.35686275]
   [0.48235294 0.4862745  0.37254903]
   [0.5058824  0.49411765 0.3882353 ]
   ...
   [0.4509804  0.45490196 0.36862746]
   [0.25882354 0.25490198 0.23137255]
   [0.10588235 0.10588235 0.10588235]]

  [[0.45490196 0.47058824 0.3529412 ]
   [0.4745098  0.47843137 0.36862746]
   [0.5058824  0.5019608  0.39607844]
   ...
   [0.45490196 0.4509804  0.36862746]
   [0.26666668 0.25490198 0.22745098]
   [0.10588235 0.10196079 0.10196079]]]]
The Validation data size is 5000
```

In [ ]:

```python
def build_encoder(input_shape):
    input_layer = Input(shape=input_shape)
    x = Conv2D(8, 3, activation='relu')(input_layer)
    print(x)
    y = MaxPool2D(2, 2)(x)
    print(y)
    z = MaxPool2D(2, 2)(y)
    print(z)
    a = Conv2D(16, 3, activation='relu')(z)
    print(a)
    b = MaxPool2D(2, 2)(a)
    print(b)
    c = Flatten()(b)
    print(c)
    d = Dense(100, activation='relu')(c)
    print(d)
    encoded = Dense(200)(d)
    print(d)
    return Model(inputs=input_layer, outputs=encoded)
input_shape = (32, 32, 3)
encoder = build_encoder(input_shape)
```

```
KerasTensor(type_spec=TensorSpec(shape=(None, 30, 30, 8), dtype=tf.float32, na
me=None), name='conv2d_2/Relu:0', description="created by layer 'conv2d_2'")
KerasTensor(type_spec=TensorSpec(shape=(None, 15, 15, 8), dtype=tf.float32, na
me=None), name='max_pooling2d_3/MaxPool:0', description="created by layer 'max
_pooling2d_3'")
KerasTensor(type_spec=TensorSpec(shape=(None, 7, 7, 8), dtype=tf.float32, name
=None), name='max_pooling2d_4/MaxPool:0', description="created by layer 'max_p
ooling2d_4'")
KerasTensor(type_spec=TensorSpec(shape=(None, 5, 5, 16), dtype=tf.float32, nam
e=None), name='conv2d_3/Relu:0', description="created by layer 'conv2d_3'")
KerasTensor(type_spec=TensorSpec(shape=(None, 2, 2, 16), dtype=tf.float32, nam
e=None), name='max_pooling2d_5/MaxPool:0', description="created by layer 'max_
pooling2d_5'")
KerasTensor(type_spec=TensorSpec(shape=(None, 64), dtype=tf.float32, name=Non
e), name='flatten_1/Reshape:0', description="created by layer 'flatten_1'")
KerasTensor(type_spec=TensorSpec(shape=(None, 100), dtype=tf.float32, name=Non
e), name='dense_3/Relu:0', description="created by layer 'dense_3'")
KerasTensor(type_spec=TensorSpec(shape=(None, 100), dtype=tf.float32, name=Non
e), name='dense_3/Relu:0', description="created by layer 'dense_3'")
```

In [ ]:
```
def build_decoder(encoded_shape):
    input_layer = Input(shape=encoded_shape)
    print (input_layer)
    x = Dense(6 * 6 * 16, activation='relu')(input_layer)
    print(x)
    y = Reshape(target_shape=(6, 6, 16))(x)
    print(y)
    z = Conv2DTranspose(8, 4, strides=2, activation='relu', output_padding=1)(y
    print(z)
    a = BatchNormalization()(z)
    print(a)
    decoded = Conv2DTranspose(3, 4, strides=2, activation='relu')(a)
    print (decoded)
    return Model(inputs=input_layer, outputs=decoded)
encoded_shape = (200,)
decoder = build_decoder(encoded_shape)
```

```
KerasTensor(type_spec=TensorSpec(shape=(None, 200), dtype=tf.float32, name='in
put_5'), name='input_5', description="created by layer 'input_5'")
KerasTensor(type_spec=TensorSpec(shape=(None, 576), dtype=tf.float32, name=Non
e), name='dense_5/Relu:0', description="created by layer 'dense_5'")
KerasTensor(type_spec=TensorSpec(shape=(None, 6, 6, 16), dtype=tf.float32, nam
e=None), name='reshape_1/Reshape:0', description="created by layer 'reshape_
1'")
KerasTensor(type_spec=TensorSpec(shape=(None, 15, 15, 8), dtype=tf.float32, na
me=None), name='conv2d_transpose_2/Relu:0', description="created by layer 'con
v2d_transpose_2'")
KerasTensor(type_spec=TensorSpec(shape=(None, 15, 15, 8), dtype=tf.float32, na
me=None), name='batch_normalization_1/FusedBatchNormV3:0', description="create
d by layer 'batch_normalization_1'")
KerasTensor(type_spec=TensorSpec(shape=(None, 32, 32, 3), dtype=tf.float32, na
me=None), name='conv2d_transpose_3/Relu:0', description="created by layer 'con
v2d_transpose_3'")
```

Connect the encoder and decoder to create the autoencoder

In [ ]:
```
input_layer = Input(shape=input_shape)
encoded = encoder(input_layer)
decoded = decoder(encoded)

autoencoder = Model(inputs=input_layer, outputs=decoded)

autoencoder.compile(optimizer='adam', loss='mae')
autoencoder.summary()
```

```
Model: "model_5"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_6 (InputLayer)        [(None, 32, 32, 3)]       0

 model_3 (Functional)        (None, 200)               28092

 model_4 (Functional)        (None, 32, 32, 3)         118251

=================================================================
Total params: 146343 (571.65 KB)
Trainable params: 146327 (571.59 KB)
Non-trainable params: 16 (64.00 Byte)
_________________________________________________________________
```

Training

In [ ]:

```
autoencoder.compile(optimizer='adam', loss='mean_squared_error')
history=autoencoder.fit(x_train, x_train, epochs=3, batch_size=10, validation_s
                validation_data=(val_data, val_data))
```

```
Epoch 1/3
5000/5000 [==============================] - 13s 3ms/step - loss: 0.0370 - val
_loss: 0.0293
Epoch 2/3
5000/5000 [==============================] - 13s 3ms/step - loss: 0.0309 - val
_loss: 0.0297
Epoch 3/3
5000/5000 [==============================] - 13s 3ms/step - loss: 0.0301 - val
_loss: 0.0282
```

In [ ]:

```
test_loss = autoencoder.evaluate(test_data, test_data)
print("Test Loss is:", test_loss)
```

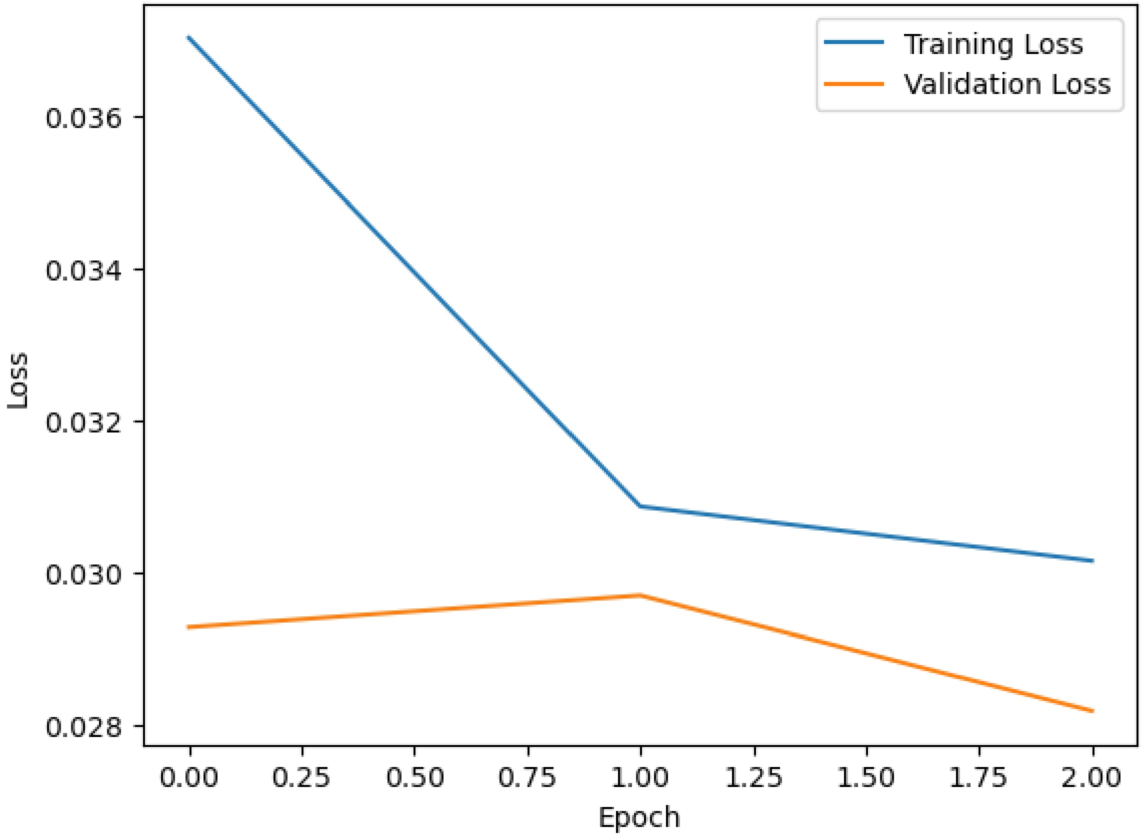
```
313/313 [==============================] - 1s 2ms/step - loss: 0.0287
Test Loss is: 0.028674405068159103
```

In [ ]:

```
import matplotlib.pyplot as plt

training_loss = history.history['loss']
validation_loss = history.history['val_loss']

# Plot the loss curves
plt.plot(training_loss, label='Training Loss')
plt.plot(validation_loss, label='Validation Loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend()
plt.show()
```



In [ ]:
```
import matplotlib.pyplot as plt

x_test = x_test

plt.figure(figsize=(22, 20))

for i in range(10):
    original_image = x_test[i]

    predicted_image = autoencoder.predict(tf.expand_dims(original_image, 0))[0]

    plt.subplot(10, 2, 2 * i + 1)
    plt.title('Original Image')
    plt.axis('on')
    plt.imshow(original_image)

    plt.subplot(10, 2, 2 * i + 2)
    plt.title('Predicted Image')
    plt.axis('on')
    plt.imshow(predicted_image)

plt.tight_layout()
plt.show()
```

```
1/1 [==============================] - 0s 94ms/step
1/1 [==============================] - 0s 6ms/step
1/1 [==============================] - 0s 6ms/step
1/1 [==============================] - 0s 6ms/step
1/1 [==============================] - 0s 6ms/step
1/1 [==============================] - 0s 6ms/step
1/1 [==============================] - 0s 7ms/step
1/1 [==============================] - 0s 6ms/step
1/1 [==============================] - 0s 6ms/step
1/1 [==============================] - 0s 6ms/step
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for fl
oats or [0..255] for integers).
```


Original Image
Predicted Image
0
10
20
30
0
20


Original Image
Predicted Image
0
10
20
30
0
20


Original Image
Predicted Image
0
10
20
30
0
20

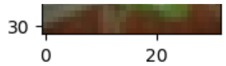

Original Image
Predicted Image
0
10
20
30
0
20

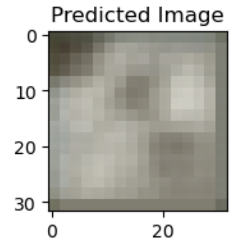

Original Image
Predicted Image
0
10
20
30
0
20

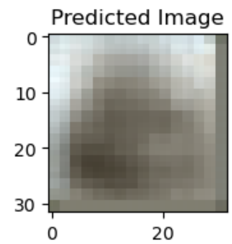

Original Image
Predicted Image
0
10
20
30
0
20


Original Image
Predicted Image
0
10
20
30
0
20


Original Image
Predicted Image
0
10
20







Autoencoder

Encoder:

Encoder is a deep learning approach where it compress the input file into a latent space representation. It takes the input image and covert it into lower dimension image of the orginal image. This compressed image is then fed as an input to the decoder.

Decoder

Decoder is a machanism which converts lower dimension as an input and generates an output, often of the same type as the input data.

Autoencoder

These neural networks consist of an encoder and decoder. They are used for tasks like data compression, feature learning, and anomaly detection. Here we take samples from Cifar10, where we train, test and validate the incoming data. Our Autoencoder has the property of encoding and decoding the images.

Testing:

We will train the autoencoder with the images from Cifar10. These imagaes are used to train our model and make model to predict and enlarge to original image

Loss Function

Choose an appropriate loss function for your task. For image prediction and enlargement, mean squared error (MSE) loss or similar reconstruction loss functions are often used. The loss measures the difference between the input image and the output image generated by the autoencoder.

In [ ]:

```

```