```
import numpy as np
import tensorflow as tf
```

```
from sklearn.metrics import classification_report, confusion_matrix,ConfusionMatrixDisplay
```

```
(x_train,y_train),(x_test,y_test)=tf.keras.datasets.cifar10.load_data()
#imports tensorflow, numpy and loads data
```

```
Downloading data from https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz
170498071/170498071 [==============================] - 15s 0us/step
```

```
#this cell one-hot encodes the labels
k=np.array(range(10))
y_train=(y_train==k).astype(int)
y_test=(y_test==k).astype(int)
```

1 Preparing Data

1. Viewing Data

- Each instance has numpy array of (32,32,3) and label ,an integer from 0 to 9
- There is 50000 training data and 10000 test data

```
type(x_train)
```

```
numpy.ndarray
```

```
x_train.shape
```

```
(50000, 32, 32, 3)
```

```
x_test.shape
```

```
(10000, 32, 32, 3)
```

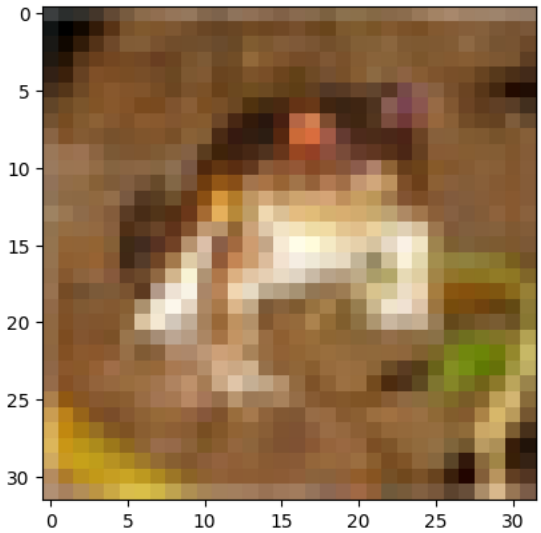
```
type(y_train),y_train.shape
```

```
(numpy.ndarray, (50000, 1))
```

```
import matplotlib.pyplot as plt
```

```
plt.imshow(x_train[0])
```

```
<matplotlib.image.AxesImage at 0x79966c71e260>
```



2 **Preprocessing data**

Converting data to tensors

```
def to_tensors(x,y):
  """
  takes x,y as numpy image and label,
  convert it to normalised tensors
  """
  newx=tf.image.convert_image_dtype(x,dtype=tf.float32)
  newy=tf.convert_to_tensor(y)
  return newx,newy
```

Data Augmentation

```
data_augment=tf.keras.Sequential([
    tf.keras.layers.RandomFlip(mode="horizontal"),
    tf.keras.layers.RandomRotation(factor=0.05)
])


from sklearn.model_selection import train_test_split
def dataaug(x,y,aug_size):
  """
  gives augmented data by random flip and random rotation,
  inputs x,y as numpy and `aug_size` as values between 0 and 1
  telling how much augmentation is needed
  :if augmentation is required to change, change it from
  `data_augment` sequential model
  """
  xtr,xte,ytr,yte=train_test_split(x,y,test_size=aug_size)
  aug_data=data_augment(xte)
  aug_data=tf.clip_by_value(aug_data,0,255)
  aug_data=aug_data/255

  yte=tf.convert_to_tensor(yte)

  return aug_data,yte
```

Converting to Datasets shuffling and Batching

```
def preprocessing(x,y,batch_size=32,aug_size=None,val_size=None):
  """
  takes x,y numpy : augments the data by aug_size (0 to 1) if specified
  and returns the batched Dataset or
  batched training and validation Dataset if val_size is specified
  """
  if aug_size!=None:
    augx,augy=dataaug(x,y,aug_size)

    x,y=to_tensors(x,y)

    x=tf.concat([x,augx],axis=0)
    y=tf.concat([y,augy],axis=0)


    data=tf.data.Dataset.from_tensor_slices((x,y))
    m=len(x)
    data=data.shuffle(buffer_size=m)


    if val_size!=None:
      val_number=int(m*val_size)
      train_size=1-val_size
      train_number=int(m*train_size)

      train_data=data.take(train_number)
      val_data=data.skip(train_number).take(val_number)

      train_data=train_data.batch(batch_size)
      val_data=val_data.batch(batch_size)

      return train_data,val_data

    else:
      return data.batch(batch_size)


  else:
    x,y=to_tensors(x,y)
    data=tf.data.Dataset.from_tensor_slices((x,y))
    m=len(x)
    data=data.shuffle(buffer_size=m)

    if val_size!=None:
      val_number=int(m*val_size)
      train_size=1-val_size
      train_number=int(m*train_size)

      train_data=data.take(train_number)
      val_data=data.skip(train_number).take(val_number)

      train_data=train_data.batch(batch_size)
      val_data=val_data.batch(batch_size)
      return train_data,val_data

    else:
      return data.batch(batch_size)



tf.__version__
```

```
'2.12.0'
```

3 **Dense Model**

```
dense_model=tf.keras.Sequential([tf.keras.layers.Flatten(),
                     tf.keras.layers.Dense(3072, activation="relu"),
                     tf.keras.layers.Dense(1024, activation="relu"),
                     tf.keras.layers.Dense(512, activation="relu"),
                     tf.keras.layers.Dense(256, activation="relu"),
                     tf.keras.layers.Dense(128, activation="relu"),
                     tf.keras.layers.Dense(64, activation="relu"),
                     tf.keras.layers.Dense(10, activation="softmax")])

dense_model.compile(optimizer='Adam',loss='categorical_crossentropy',metrics=['accuracy'])


data=preprocessing(x_train,y_train)
```

```
history=dense_model.fit(data,epochs=30)
```

```
Epoch 1/30
1563/1563 [==============================] - 18s 6ms/step - loss: 1.9616 - accuracy: 0.2724
Epoch 2/30
1563/1563 [==============================] - 10s 6ms/step - loss: 1.7559 - accuracy: 0.3637
Epoch 3/30
1563/1563 [==============================] - 11s 7ms/step - loss: 1.6657 - accuracy: 0.4012
Epoch 4/30
1563/1563 [==============================] - 10s 6ms/step - loss: 1.6015 - accuracy: 0.4256
Epoch 5/30
1563/1563 [==============================] - 10s 6ms/step - loss: 1.5530 - accuracy: 0.4426
Epoch 6/30
1563/1563 [==============================] - 9s 6ms/step - loss: 1.5122 - accuracy: 0.4586
Epoch 7/30
1563/1563 [==============================] - 11s 7ms/step - loss: 1.4788 - accuracy: 0.4709
Epoch 8/30
1563/1563 [==============================] - 10s 6ms/step - loss: 1.4442 - accuracy: 0.4855
Epoch 9/30
1563/1563 [==============================] - 10s 6ms/step - loss: 1.4185 - accuracy: 0.4933
Epoch 10/30
1563/1563 [==============================] - 10s 6ms/step - loss: 1.3909 - accuracy: 0.5027
Epoch 11/30
1563/1563 [==============================] - 13s 8ms/step - loss: 1.3638 - accuracy: 0.5123
Epoch 12/30
1563/1563 [==============================] - 12s 7ms/step - loss: 1.3370 - accuracy: 0.5230
Epoch 13/30
1563/1563 [==============================] - 10s 6ms/step - loss: 1.3189 - accuracy: 0.5285
Epoch 14/30
1563/1563 [==============================] - 10s 6ms/step - loss: 1.2888 - accuracy: 0.5362
Epoch 15/30
1563/1563 [==============================] - 9s 6ms/step - loss: 1.2680 - accuracy: 0.5472
Epoch 16/30
1563/1563 [==============================] - 10s 6ms/step - loss: 1.2403 - accuracy: 0.5547
Epoch 17/30
1563/1563 [==============================] - 10s 6ms/step - loss: 1.2203 - accuracy: 0.5614
Epoch 18/30
1563/1563 [==============================] - 9s 6ms/step - loss: 1.1947 - accuracy: 0.5708
Epoch 19/30
1563/1563 [==============================] - 9s 6ms/step - loss: 1.1702 - accuracy: 0.5763
Epoch 20/30
1563/1563 [==============================] - 10s 6ms/step - loss: 1.1423 - accuracy: 0.5907
Epoch 21/30
1563/1563 [==============================] - 10s 6ms/step - loss: 1.1221 - accuracy: 0.5946
Epoch 22/30
1563/1563 [==============================] - 10s 6ms/step - loss: 1.0942 - accuracy: 0.6058
Epoch 23/30
1563/1563 [==============================] - 10s 6ms/step - loss: 1.0729 - accuracy: 0.6134
Epoch 24/30
1563/1563 [==============================] - 10s 6ms/step - loss: 1.0463 - accuracy: 0.6201
Epoch 25/30
1563/1563 [==============================] - 10s 6ms/step - loss: 1.0254 - accuracy: 0.6286
Epoch 26/30
1563/1563 [==============================] - 9s 6ms/step - loss: 0.9976 - accuracy: 0.6367
Epoch 27/30
1563/1563 [==============================] - 10s 6ms/step - loss: 0.9724 - accuracy: 0.6467
Epoch 28/30
1563/1563 [==============================] - 10s 6ms/step - loss: 0.9530 - accuracy: 0.6556
Epoch 29/30
1563/1563 [==============================] - 10s 6ms/step - loss: 0.9306 - accuracy: 0.6655
```

```
a,b=to_tensors(x_test,y_test)
dense_model.evaluate(a,b)
```

```
313/313 [==============================] - 1s 3ms/step - loss: 1.7679 - accuracy: 0.4827
[1.7678788900375366, 0.482699990272522]
```

```
dense_model.save("drive/MyDrive/Colab Notebooks/Cifar10/saves/dense_model",save_format='tf')
```

```
WARNING:absl:Found untraced functions such as _update_step_xla while saving (showing 1 of 1). These functions will not be directly c
```

```
dense_model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 flatten (Flatten)           (None, 3072)              0

 dense (Dense)               (None, 3072)              9440256

 dense_1 (Dense)             (None, 1024)              3146752

 dense_2 (Dense)             (None, 512)               524800
```

```
 dense_3 (Dense)             (None, 256)               131328

 dense_4 (Dense)             (None, 128)               32896

 dense_5 (Dense)             (None, 64)                8256

 dense_6 (Dense)             (None, 10)                650

=================================================================
Total params: 13,284,938
Trainable params: 13,284,938
Non-trainable params: 0
_________________________________________________________________
```

```
a,b=to_tensors(x_test,y_test)
y_preds=dense_model.predict(a)
y_preds=[np.argmax(x) for x in y_preds]
y_true=[np.argmax(x) for x in y_test]
```

```
313/313 [==============================] - 1s 2ms/step
```

```
print(classification_report(y_preds,y_true))
```

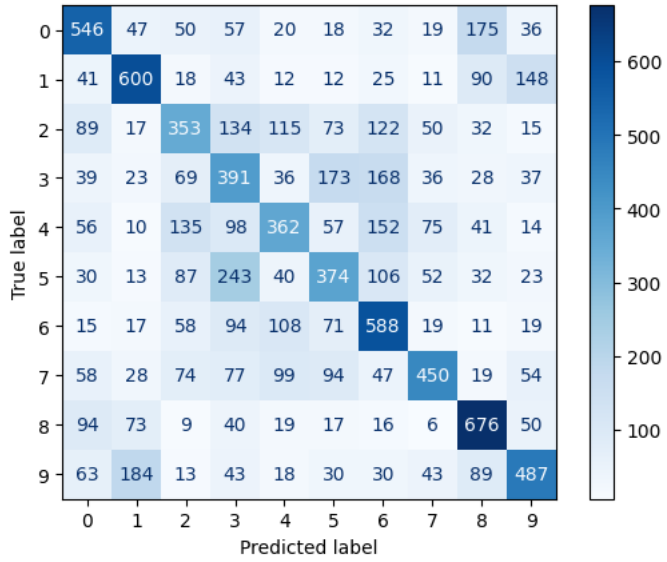
```
              precision    recall  f1-score   support

           0       0.55      0.53      0.54      1031
           1       0.60      0.59      0.60      1012
           2       0.35      0.41      0.38       866
           3       0.39      0.32      0.35      1220
           4       0.36      0.44      0.40       829
           5       0.37      0.41      0.39       919
           6       0.59      0.46      0.51      1286
           7       0.45      0.59      0.51       761
           8       0.68      0.57      0.62      1193
           9       0.49      0.55      0.52       883

    accuracy                           0.48     10000
   macro avg       0.48      0.49      0.48     10000
weighted avg       0.49      0.48      0.48     10000
```

```
print(confusion_matrix(y_preds,y_true))
```

```
[[546  41  89  39  56  30  15  58  94  63]
 [ 47 600  17  23  10  13  17  28  73 184]
 [ 50  18 353  69 135  87  58  74   9  13]
 [ 57  43 134 391  98 243  94  77  40  43]
 [ 20  12 115  36 362  40 108  99  19  18]
 [ 18  12  73 173  57 374  71  94  17  30]
 [ 32  25 122 168 152 106 588  47  16  30]
 [ 19  11  50  36  75  52  19 450   6  43]
 [175  90  32  28  41  32  11  19 676  89]
 [ 36 148  15  37  14  23  19  54  50 487]]
```

```
ConfusionMatrixDisplay.from_predictions(y_true=y_true,y_pred=y_preds,cmap='Blues')
```

```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7aa590339000>
```



3 **Custom CNN Model**

Model

```
#the model is custom designed, by referring multiple architecture Like VGG-16, LeNet-5

model_1=tf.keras.Sequential(
                      [tf.keras.layers.Conv2D(filters=32,kernel_size=5,input_shape=[32,32,3],padding='same'),
                      tf.keras.layers.Activation('relu'),
                      tf.keras.layers.MaxPool2D(),

                      tf.keras.layers.Conv2D(filters=64,kernel_size=3,padding='same'),
                      tf.keras.layers.Activation('relu'),
                      tf.keras.layers.MaxPool2D(),


                      tf.keras.layers.Conv2D(filters=64,kernel_size=3,padding='same'),


                      tf.keras.layers.Conv2D(kernel_size=3,filters=128,padding='same'),
                      tf.keras.layers.Activation('relu'),
                      tf.keras.layers.MaxPool2D(),


                      tf.keras.layers.Conv2D(filters=256,kernel_size=3,padding='same'),
                      tf.keras.layers.Activation('relu'),
                      tf.keras.layers.MaxPool2D(),

                      tf.keras.layers.Conv2D(kernel_size=3,filters=256,padding='same'),
                      tf.keras.layers.Activation('relu'),
                      tf.keras.layers.MaxPool2D(),

                      tf.keras.layers.Flatten(),
                      tf.keras.layers.Dense(128,activation='relu'),
                      tf.keras.layers.Dense(64,activation='relu'),
                      tf.keras.layers.Dense(10,activation='softmax')
                       ]
                      )
model_1.compile(optimizer='Adam',loss='categorical_crossentropy',metrics=['accuracy'])


data=preprocessing(x_train,y_train)


history=model_1.fit(data,epochs=30)
```

```
Epoch 1/30
1563/1563 [==============================] - 22s 6ms/step - loss: 1.6982 - accuracy: 0.3655
Epoch 2/30
1563/1563 [==============================] - 10s 6ms/step - loss: 1.2528 - accuracy: 0.5489
Epoch 3/30
1563/1563 [==============================] - 9s 6ms/step - loss: 1.0631 - accuracy: 0.6226
Epoch 4/30
1563/1563 [==============================] - 10s 6ms/step - loss: 0.9257 - accuracy: 0.6745
Epoch 5/30
1563/1563 [==============================] - 10s 6ms/step - loss: 0.8247 - accuracy: 0.7114
Epoch 6/30
1563/1563 [==============================] - 9s 6ms/step - loss: 0.7437 - accuracy: 0.7412
Epoch 7/30
1563/1563 [==============================] - 10s 6ms/step - loss: 0.6686 - accuracy: 0.7656
Epoch 8/30
1563/1563 [==============================] - 10s 6ms/step - loss: 0.6101 - accuracy: 0.7875
Epoch 9/30
1563/1563 [==============================] - 10s 6ms/step - loss: 0.5461 - accuracy: 0.8084
Epoch 10/30
1563/1563 [==============================] - 9s 6ms/step - loss: 0.5068 - accuracy: 0.8244
Epoch 11/30
1563/1563 [==============================] - 10s 6ms/step - loss: 0.4548 - accuracy: 0.8420
Epoch 12/30
1563/1563 [==============================] - 10s 6ms/step - loss: 0.4158 - accuracy: 0.8548
Epoch 13/30
1563/1563 [==============================] - 9s 6ms/step - loss: 0.3717 - accuracy: 0.8704
Epoch 14/30
1563/1563 [==============================] - 9s 6ms/step - loss: 0.3518 - accuracy: 0.8781
Epoch 15/30
1563/1563 [==============================] - 10s 6ms/step - loss: 0.3162 - accuracy: 0.8904
Epoch 16/30
1563/1563 [==============================] - 10s 6ms/step - loss: 0.2927 - accuracy: 0.8999
Epoch 17/30
1563/1563 [==============================] - 9s 6ms/step - loss: 0.2768 - accuracy: 0.9058
```

```
Epoch 18/30
1563/1563 [==============================] - 10s 6ms/step - loss: 0.2562 - accuracy: 0.9126
Epoch 19/30
1563/1563 [==============================] - 10s 6ms/step - loss: 0.2418 - accuracy: 0.9171
Epoch 20/30
1563/1563 [==============================] - 9s 6ms/step - loss: 0.2285 - accuracy: 0.9233
Epoch 21/30
1563/1563 [==============================] - 9s 6ms/step - loss: 0.2183 - accuracy: 0.9273
Epoch 22/30
1563/1563 [==============================] - 10s 6ms/step - loss: 0.2000 - accuracy: 0.9332
Epoch 23/30
1563/1563 [==============================] - 10s 6ms/step - loss: 0.1975 - accuracy: 0.9356
Epoch 24/30
1563/1563 [==============================] - 9s 6ms/step - loss: 0.1899 - accuracy: 0.9382
Epoch 25/30
1563/1563 [==============================] - 10s 6ms/step - loss: 0.1797 - accuracy: 0.9419
Epoch 26/30
1563/1563 [==============================] - 10s 6ms/step - loss: 0.1739 - accuracy: 0.9448
Epoch 27/30
1563/1563 [==============================] - 10s 6ms/step - loss: 0.1693 - accuracy: 0.9459
Epoch 28/30
1563/1563 [==============================] - 9s 6ms/step - loss: 0.1625 - accuracy: 0.9466
Epoch 29/30
```

```
a,b=to_tensors(x_test,y_test)
model_1.evaluate(a,b)
```

```
313/313 [==============================] - 1s 4ms/step - loss: 1.4329 - accuracy: 0.7143
[1.4328725337982178, 0.7142999768257141]
```

```
model_1.summary()
```

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 32, 32, 32)        2432

 activation (Activation)     (None, 32, 32, 32)        0

 max_pooling2d (MaxPooling2D  (None, 16, 16, 32)       0
 )

 conv2d_1 (Conv2D)           (None, 16, 16, 64)        18496

 activation_1 (Activation)   (None, 16, 16, 64)        0

 max_pooling2d_1 (MaxPooling  (None, 8, 8, 64)         0
 2D)

 conv2d_2 (Conv2D)           (None, 8, 8, 64)          36928

 conv2d_3 (Conv2D)           (None, 8, 8, 128)         73856

 activation_2 (Activation)   (None, 8, 8, 128)         0

 max_pooling2d_2 (MaxPooling  (None, 4, 4, 128)        0
 2D)

 conv2d_4 (Conv2D)           (None, 4, 4, 256)         295168

 activation_3 (Activation)   (None, 4, 4, 256)         0

 max_pooling2d_3 (MaxPooling  (None, 2, 2, 256)        0
 2D)

 conv2d_5 (Conv2D)           (None, 2, 2, 256)         590080

 activation_4 (Activation)   (None, 2, 2, 256)         0

 max_pooling2d_4 (MaxPooling  (None, 1, 1, 256)        0
 2D)

 flatten_1 (Flatten)         (None, 256)               0

 dense_7 (Dense)             (None, 128)               32896

 dense_8 (Dense)             (None, 64)                8256

 dense_9 (Dense)             (None, 10)                650

=================================================================
Total params: 1,058,762
Trainable params: 1,058,762
Non-trainable params: 0
_________________________________________________________________
```

```
history=model_1.fit(data,epochs=30)
```

```
Epoch 2/30
1563/1563 [==============================] - 12s 8ms/step - loss: 1.1898 - accuracy: 0.5729
Epoch 3/30
1563/1563 [==============================] - 10s 6ms/step - loss: 0.9818 - accuracy: 0.6519
Epoch 4/30
1563/1563 [==============================] - 16s 10ms/step - loss: 0.8430 - accuracy: 0.7057
Epoch 5/30
1563/1563 [==============================] - 10s 6ms/step - loss: 0.7387 - accuracy: 0.7424
Epoch 6/30
1563/1563 [==============================] - 10s 6ms/step - loss: 0.6660 - accuracy: 0.7679
Epoch 7/30
1563/1563 [==============================] - 10s 6ms/step - loss: 0.5805 - accuracy: 0.7974
Epoch 8/30
1563/1563 [==============================] - 10s 6ms/step - loss: 0.5245 - accuracy: 0.8189
Epoch 9/30
1563/1563 [==============================] - 9s 6ms/step - loss: 0.4680 - accuracy: 0.8364
Epoch 10/30
1563/1563 [==============================] - 10s 6ms/step - loss: 0.4206 - accuracy: 0.8532
Epoch 11/30
1563/1563 [==============================] - 11s 7ms/step - loss: 0.3785 - accuracy: 0.8685
Epoch 12/30
1563/1563 [==============================] - 11s 7ms/step - loss: 0.3450 - accuracy: 0.8807
Epoch 13/30
1563/1563 [==============================] - 11s 7ms/step - loss: 0.3095 - accuracy: 0.8939
Epoch 14/30
1563/1563 [==============================] - 11s 7ms/step - loss: 0.2773 - accuracy: 0.9049
Epoch 15/30
1563/1563 [==============================] - 11s 7ms/step - loss: 0.2601 - accuracy: 0.9100
Epoch 16/30
1563/1563 [==============================] - 10s 7ms/step - loss: 0.2492 - accuracy: 0.9157
Epoch 17/30
1563/1563 [==============================] - 10s 6ms/step - loss: 0.2249 - accuracy: 0.9255
Epoch 18/30
1563/1563 [==============================] - 10s 7ms/step - loss: 0.2044 - accuracy: 0.9307
Epoch 19/30
1563/1563 [==============================] - 10s 7ms/step - loss: 0.1984 - accuracy: 0.9332
Epoch 20/30
1563/1563 [==============================] - 10s 6ms/step - loss: 0.1958 - accuracy: 0.9361
Epoch 21/30
1563/1563 [==============================] - 11s 7ms/step - loss: 0.1902 - accuracy: 0.9372
Epoch 22/30
1563/1563 [==============================] - 10s 7ms/step - loss: 0.1831 - accuracy: 0.9407
Epoch 23/30
1563/1563 [==============================] - 11s 7ms/step - loss: 0.1613 - accuracy: 0.9464
Epoch 24/30
1563/1563 [==============================] - 10s 7ms/step - loss: 0.1628 - accuracy: 0.9472
Epoch 25/30
1563/1563 [==============================] - 11s 7ms/step - loss: 0.1564 - accuracy: 0.9496
Epoch 26/30
1563/1563 [==============================] - 11s 7ms/step - loss: 0.1537 - accuracy: 0.9516
Epoch 27/30
1563/1563 [==============================] - 10s 6ms/step - loss: 0.1448 - accuracy: 0.9541
Epoch 28/30
1563/1563 [==============================] - 10s 7ms/step - loss: 0.1362 - accuracy: 0.9566
Epoch 29/30
1563/1563 [==============================] - 11s 7ms/step - loss: 0.1599 - accuracy: 0.9509
Epoch 30/30
1563/1563 [==============================] - 10s 7ms/step - loss: 0.1328 - accuracy: 0.9604
```

```
a,b=to_tensors(x_test,y_test)
y_preds=model_1.predict(a)
y_preds=[np.argmax(x) for x in y_preds]
y_true=[np.argmax(x) for x in y_test]
```

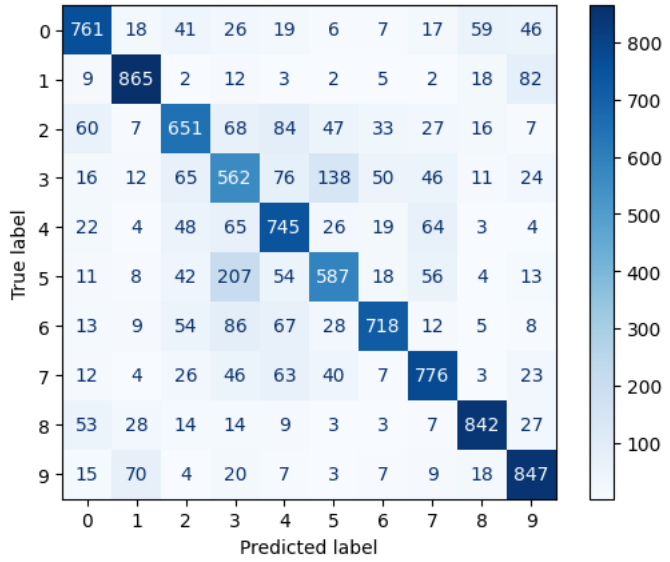
```
313/313 [==============================] - 1s 2ms/step
```

```
print(classification_report(y_preds,y_true))
```

```
              precision    recall  f1-score   support

           0       0.76      0.78      0.77       972
           1       0.86      0.84      0.85      1025
           2       0.65      0.69      0.67       947
           3       0.56      0.51      0.53      1106
           4       0.74      0.66      0.70      1127
           5       0.59      0.67      0.62       880
           6       0.72      0.83      0.77       867
           7       0.78      0.76      0.77      1016
           8       0.84      0.86      0.85       979
           9       0.85      0.78      0.81      1081

    accuracy                           0.74     10000
   macro avg       0.74      0.74      0.74     10000
weighted avg       0.74      0.74      0.74     10000
```

```
ConfusionMatrixDisplay.from_predictions(y_true=y_true,y_pred=y_preds,cmap='Blues')
```

```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7c4b33986aa0>
```



Final Main Model

```
#the model is custom designed, by referring multiple architecture Like VGG-16, LeNet-5
#have implemented increasing dropout for regularisation and BatchNormalisation
model=tf.keras.Sequential(
                      [tf.keras.layers.Conv2D(filters=32,kernel_size=5,input_shape=[32,32,3],padding='same'),
                      tf.keras.layers.BatchNormalization(),
                      tf.keras.layers.Activation('relu'),
                      tf.keras.layers.MaxPool2D(),

                      tf.keras.layers.Conv2D(filters=64,kernel_size=3,padding='same'),
                      tf.keras.layers.BatchNormalization(),
                      tf.keras.layers.Activation('relu'),
                      tf.keras.layers.MaxPool2D(),
                      tf.keras.layers.Dropout(0.2),

                      tf.keras.layers.Conv2D(filters=64,kernel_size=3,padding='same'),


                      tf.keras.layers.Conv2D(kernel_size=3,filters=128,padding='same'),
                      tf.keras.layers.BatchNormalization(),
                      tf.keras.layers.Activation('relu'),
                      tf.keras.layers.MaxPool2D(),
                      tf.keras.layers.Dropout(0.3),

                      tf.keras.layers.Conv2D(filters=256,kernel_size=3,padding='same'),
                      tf.keras.layers.BatchNormalization(),
                      tf.keras.layers.Activation('relu'),
                      tf.keras.layers.MaxPool2D(),
                      tf.keras.layers.Dropout(0.3),

                      tf.keras.layers.Conv2D(kernel_size=3,filters=256,padding='same'),
                      tf.keras.layers.BatchNormalization(),
                      tf.keras.layers.Activation('relu'),
                      tf.keras.layers.MaxPool2D(),

                      tf.keras.layers.Flatten(),
                      tf.keras.layers.Dense(128,activation='relu'),
                      tf.keras.layers.Dropout(0.4),
                      tf.keras.layers.Dense(64,activation='relu'),
                      tf.keras.layers.Dense(10,activation='softmax')
                       ]
                      )
model.compile(optimizer='Adam',loss='categorical_crossentropy',metrics=['accuracy'])


model.summary()
```

```
conv2d_8 (Conv2D)           (None, 8, 8, 64)          36928

conv2d_9 (Conv2D)           (None, 8, 8, 128)         73856

batch_normalization_2 (Batc  (None, 8, 8, 128)        512
hNormalization)

activation_7 (Activation)   (None, 8, 8, 128)         0

max_pooling2d_7 (MaxPooling  (None, 4, 4, 128)        0
2D)

dropout_1 (Dropout)         (None, 4, 4, 128)         0

conv2d_10 (Conv2D)          (None, 4, 4, 256)         295168

batch_normalization_3 (Batc  (None, 4, 4, 256)        1024
hNormalization)

activation_8 (Activation)   (None, 4, 4, 256)         0

max_pooling2d_8 (MaxPooling  (None, 2, 2, 256)        0
2D)

dropout_2 (Dropout)         (None, 2, 2, 256)         0

conv2d_11 (Conv2D)          (None, 2, 2, 256)         590080

batch_normalization_4 (Batc  (None, 2, 2, 256)        1024
hNormalization)

activation_9 (Activation)   (None, 2, 2, 256)         0

max_pooling2d_9 (MaxPooling  (None, 1, 1, 256)        0
2D)

flatten_1 (Flatten)         (None, 256)               0

dense_3 (Dense)             (None, 128)               32896

dropout_3 (Dropout)         (None, 128)               0

dense_4 (Dense)             (None, 64)                8256

dense_5 (Dense)             (None, 10)                650

=================================================================
Total params: 1,061,706
Trainable params: 1,060,234
Non-trainable params: 1,472
_________________________________________________________________
```

Callbacks

```
def create_earlystopping():
  return tf.keras.callbacks.EarlyStopping(monitor="val_loss",patience=10)


import datetime
import os
def create_tensorboard():
  logs=os.path.join('drive/MyDrive/Colab Notebooks/Cifar10/logtb',datetime.datetime.now().strftime("%d_%m_%Y-%H_%M_%S"))
  return tf.keras.callbacks.TensorBoard(log_dir=logs)


def create_checkpoints():
  logs=os.path.join('drive/MyDrive/Colab Notebooks/Cifar10/saves/withvalid',datetime.datetime.now().strftime("%d_%m_%Y-%H_%M_%S"))
  logs=os.path.join(logs,'save.h5')
  return tf.keras.callbacks.ModelCheckpoint(filepath=logs,save_best_only=True,save_weights_only=True,verbose=1)
  #,save_best_only=True
```

Training

With validation set

```
tb=create_tensorboard()
sav=create_checkpoints()
earlystop=create_earlystopping()
```

```
train,valid=preprocessing(x_train,y_train,aug_size=0.99,val_size=0.15)
```

while training with validation set here, the validation accuracy is higher than training set accuracy, and it is mainly because validation set contains subset of data_augmentation, in which some images might have got few augmentation, resulting similar datas in training as well as validation set.

```
history=model.fit(train,epochs=300,validation_data=valid,callbacks=[tb,sav,earlystop])
```

```
Epoch 1/300
2643/2643 [==============================] - ETA: 0s - loss: 1.5561 - accuracy: 0.4301
Epoch 1: val_loss improved from inf to 1.16772, saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/withvalid/19_07_2023-
2643/2643 [==============================] - 44s 10ms/step - loss: 1.5561 - accuracy: 0.4301 - val_loss: 1.1677 - val_accuracy: 0
Epoch 2/300
2641/2643 [============================>.] - ETA: 0s - loss: 1.2038 - accuracy: 0.5743
Epoch 2: val_loss did not improve from 1.16772
2643/2643 [==============================] - 26s 10ms/step - loss: 1.2039 - accuracy: 0.5743 - val_loss: 1.2379 - val_accuracy: 0
Epoch 3/300
2638/2643 [============================>.] - ETA: 0s - loss: 1.0435 - accuracy: 0.6379
Epoch 3: val_loss improved from 1.16772 to 0.98975, saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/withvalid/19_07_2
2643/2643 [==============================] - 26s 10ms/step - loss: 1.0433 - accuracy: 0.6379 - val_loss: 0.9897 - val_accuracy: 0
Epoch 4/300
2643/2643 [==============================] - ETA: 0s - loss: 0.9340 - accuracy: 0.6812
Epoch 4: val_loss improved from 0.98975 to 0.98444, saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/withvalid/19_07_2
2643/2643 [==============================] - 26s 10ms/step - loss: 0.9340 - accuracy: 0.6812 - val_loss: 0.9844 - val_accuracy: 0
Epoch 5/300
2643/2643 [==============================] - ETA: 0s - loss: 0.8683 - accuracy: 0.7066
Epoch 5: val_loss improved from 0.98444 to 0.84699, saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/withvalid/19_07_2
2643/2643 [==============================] - 25s 10ms/step - loss: 0.8683 - accuracy: 0.7066 - val_loss: 0.8470 - val_accuracy: 0
Epoch 6/300
2642/2643 [============================>.] - ETA: 0s - loss: 0.8087 - accuracy: 0.7272
Epoch 6: val_loss improved from 0.84699 to 0.72317, saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/withvalid/19_07_2
2643/2643 [==============================] - 26s 10ms/step - loss: 0.8087 - accuracy: 0.7272 - val_loss: 0.7232 - val_accuracy: 0
Epoch 7/300
2642/2643 [============================>.] - ETA: 0s - loss: 0.7674 - accuracy: 0.7428
Epoch 7: val_loss improved from 0.72317 to 0.64327, saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/withvalid/19_07_2
2643/2643 [==============================] - 26s 10ms/step - loss: 0.7674 - accuracy: 0.7429 - val_loss: 0.6433 - val_accuracy: 0
Epoch 8/300
2641/2643 [============================>.] - ETA: 0s - loss: 0.7225 - accuracy: 0.7573
Epoch 8: val_loss did not improve from 0.64327
2643/2643 [==============================] - 25s 10ms/step - loss: 0.7224 - accuracy: 0.7573 - val_loss: 0.6631 - val_accuracy: 0
Epoch 9/300
2642/2643 [============================>.] - ETA: 0s - loss: 0.6905 - accuracy: 0.7685
Epoch 9: val_loss improved from 0.64327 to 0.63654, saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/withvalid/19_07_2
2643/2643 [==============================] - 27s 10ms/step - loss: 0.6904 - accuracy: 0.7685 - val_loss: 0.6365 - val_accuracy: 0
Epoch 10/300
2642/2643 [============================>.] - ETA: 0s - loss: 0.6572 - accuracy: 0.7794
Epoch 10: val_loss did not improve from 0.63654
2643/2643 [==============================] - 25s 10ms/step - loss: 0.6573 - accuracy: 0.7794 - val_loss: 0.7528 - val_accuracy: 0
Epoch 11/300
2640/2643 [============================>.] - ETA: 0s - loss: 0.6350 - accuracy: 0.7874
Epoch 11: val_loss did not improve from 0.63654
2643/2643 [==============================] - 25s 9ms/step - loss: 0.6353 - accuracy: 0.7873 - val_loss: 0.7074 - val_accuracy: 0.
Epoch 12/300
2642/2643 [============================>.] - ETA: 0s - loss: 0.6169 - accuracy: 0.7937
Epoch 12: val_loss improved from 0.63654 to 0.59328, saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/withvalid/19_07_
2643/2643 [==============================] - 26s 10ms/step - loss: 0.6168 - accuracy: 0.7937 - val_loss: 0.5933 - val_accuracy: 0
Epoch 13/300
2643/2643 [==============================] - ETA: 0s - loss: 0.5957 - accuracy: 0.7989
Epoch 13: val_loss improved from 0.59328 to 0.52082, saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/withvalid/19_07_
2643/2643 [==============================] - 25s 10ms/step - loss: 0.5957 - accuracy: 0.7989 - val_loss: 0.5208 - val_accuracy: 0
Epoch 14/300
2642/2643 [============================>.] - ETA: 0s - loss: 0.5739 - accuracy: 0.8066
Epoch 14: val_loss did not improve from 0.52082
2643/2643 [==============================] - 25s 10ms/step - loss: 0.5739 - accuracy: 0.8066 - val_loss: 0.6762 - val_accuracy: 0
Epoch 15/300
```

```
history2=model.fit(train,epochs=300,validation_data=valid,callbacks=[tb,sav,earlystop])
```

```
Epoch 1/300
2641/2643 [============================>.] - ETA: 0s - loss: 0.4732 - accuracy: 0.8386
Epoch 1: val_loss improved from inf to 0.39217, saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/withvalid/19_07_2023-
2643/2643 [==============================] - 25s 9ms/step - loss: 0.4731 - accuracy: 0.8386 - val_loss: 0.3922 - val_accuracy: 0.
Epoch 2/300
2642/2643 [============================>.] - ETA: 0s - loss: 0.4667 - accuracy: 0.8429
Epoch 2: val_loss did not improve from 0.39217
2643/2643 [==============================] - 25s 9ms/step - loss: 0.4667 - accuracy: 0.8429 - val_loss: 0.4424 - val_accuracy: 0.
Epoch 3/300
2642/2643 [============================>.] - ETA: 0s - loss: 0.4520 - accuracy: 0.8486
Epoch 3: val_loss improved from 0.39217 to 0.35738, saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/withvalid/19_07_2
2643/2643 [==============================] - 26s 10ms/step - loss: 0.4519 - accuracy: 0.8485 - val_loss: 0.3574 - val_accuracy: 0
Epoch 4/300
2642/2643 [============================>.] - ETA: 0s - loss: 0.4512 - accuracy: 0.8468
Epoch 4: val_loss did not improve from 0.35738
2643/2643 [==============================] - 25s 9ms/step - loss: 0.4511 - accuracy: 0.8469 - val_loss: 0.4324 - val_accuracy: 0.
Epoch 5/300
```

```
2642/2643 [============================>.] - ETA: 0s - loss: 0.4420 - accuracy: 0.8498
Epoch 5: val_loss did not improve from 0.35738
2643/2643 [==============================] - 26s 10ms/step - loss: 0.4420 - accuracy: 0.8498 - val_loss: 0.4817 - val_accuracy: 0
Epoch 6/300
2642/2643 [============================>.] - ETA: 0s - loss: 0.4359 - accuracy: 0.8534
Epoch 6: val_loss improved from 0.35738 to 0.35658, saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/withvalid/19_07_2
2643/2643 [==============================] - 26s 10ms/step - loss: 0.4359 - accuracy: 0.8533 - val_loss: 0.3566 - val_accuracy: 0
Epoch 7/300
2643/2643 [==============================] - ETA: 0s - loss: 0.4305 - accuracy: 0.8568
Epoch 7: val_loss did not improve from 0.35658
2643/2643 [==============================] - 24s 9ms/step - loss: 0.4305 - accuracy: 0.8568 - val_loss: 0.5276 - val_accuracy: 0.
Epoch 8/300
2639/2643 [============================>.] - ETA: 0s - loss: 0.4166 - accuracy: 0.8583
Epoch 8: val_loss did not improve from 0.35658
2643/2643 [==============================] - 25s 9ms/step - loss: 0.4165 - accuracy: 0.8584 - val_loss: 0.6167 - val_accuracy: 0.
Epoch 9/300
2639/2643 [============================>.] - ETA: 0s - loss: 0.4123 - accuracy: 0.8608
Epoch 9: val_loss improved from 0.35658 to 0.32850, saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/withvalid/19_07_2
2643/2643 [==============================] - 27s 10ms/step - loss: 0.4122 - accuracy: 0.8609 - val_loss: 0.3285 - val_accuracy: 0
Epoch 10/300
2642/2643 [============================>.] - ETA: 0s - loss: 0.4053 - accuracy: 0.8619
Epoch 10: val_loss did not improve from 0.32850
2643/2643 [==============================] - 25s 9ms/step - loss: 0.4053 - accuracy: 0.8620 - val_loss: 0.4035 - val_accuracy: 0.
Epoch 11/300
2639/2643 [============================>.] - ETA: 0s - loss: 0.4015 - accuracy: 0.8638
Epoch 11: val_loss improved from 0.32850 to 0.30514, saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/withvalid/19_07_
2643/2643 [==============================] - 26s 10ms/step - loss: 0.4015 - accuracy: 0.8638 - val_loss: 0.3051 - val_accuracy: 0
Epoch 12/300
2642/2643 [============================>.] - ETA: 0s - loss: 0.3996 - accuracy: 0.8651
Epoch 12: val_loss did not improve from 0.30514
2643/2643 [==============================] - 25s 9ms/step - loss: 0.3996 - accuracy: 0.8651 - val_loss: 0.3054 - val_accuracy: 0.
Epoch 13/300
2642/2643 [============================>.] - ETA: 0s - loss: 0.3917 - accuracy: 0.8667
Epoch 13: val_loss improved from 0.30514 to 0.27189, saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/withvalid/19_07_
2643/2643 [==============================] - 26s 10ms/step - loss: 0.3917 - accuracy: 0.8667 - val_loss: 0.2719 - val_accuracy: 0
Epoch 14/300
2642/2643 [============================>.] - ETA: 0s - loss: 0.3820 - accuracy: 0.8696
Epoch 14: val_loss improved from 0.27189 to 0.25592, saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/withvalid/19_07_
2643/2643 [==============================] - 26s 10ms/step - loss: 0.3820 - accuracy: 0.8696 - val_loss: 0.2559 - val_accuracy: 0
```

```
# saving the model
model.save("drive/MyDrive/Colab Notebooks/Cifar10/saves/with_valid/model2",save_format='tf')
```

```
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_
```

```
model=tf.keras.models.load_model("drive/MyDrive/Colab Notebooks/Cifar10/saves/with_valid/model2")
```

```
a,b=to_tensors(x_test,y_test)
model.evaluate(a,b)
```

```
313/313 [==============================] - 1s 4ms/step - loss: 0.4604 - accuracy: 0.8500
[0.4603598117828369, 0.8500000238418579]
```

Accuracy in test data set is 85%

Without validation set

```
newdata=preprocessing(x_train,y_train,aug_size=0.99)
```

```
len(newdata)
```

```
3110
```

```
# stopped training early manually by 100 epochs to prevent overfitting
model.fit(newdata,epochs=200,callbacks=[sav])
```

```
Epoch 1/200
3110/3110 [==============================] - ETA: 0s - loss: 1.4992 - accuracy: 0.454
Epoch 1: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_36
3110/3110 [==============================] - 52s 9ms/step - loss: 1.4992 - accuracy:
Epoch 2/200
3105/3110 [============================>.] - ETA: 0s - loss: 1.1403 - accuracy: 0.602
Epoch 2: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_36
3110/3110 [==============================] - 29s 9ms/step - loss: 1.1402 - accuracy:
Epoch 3/200
3110/3110 [==============================] - ETA: 0s - loss: 0.9777 - accuracy: 0.665
Epoch 3: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_36
3110/3110 [==============================] - 30s 10ms/step - loss: 0.9777 - accuracy:
Epoch 4/200
3105/3110 [============================>.] - ETA: 0s - loss: 0.8796 - accuracy: 0.702
Epoch 4: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_36
3110/3110 [==============================] - 30s 10ms/step - loss: 0.8794 - accuracy:
Epoch 5/200
3104/3110 [============================>.] - ETA: 0s - loss: 0.8138 - accuracy: 0.727
Epoch 5: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_36
3110/3110 [==============================] - 29s 9ms/step - loss: 0.8138 - accuracy:
Epoch 6/200
3109/3110 [============================>.] - ETA: 0s - loss: 0.7540 - accuracy: 0.747
Epoch 6: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_36
3110/3110 [==============================] - 29s 9ms/step - loss: 0.7540 - accuracy:
Epoch 7/200
3107/3110 [============================>.] - ETA: 0s - loss: 0.7073 - accuracy: 0.762
Epoch 7: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_36
3110/3110 [==============================] - 28s 9ms/step - loss: 0.7073 - accuracy:
Epoch 8/200
3109/3110 [============================>.] - ETA: 0s - loss: 0.6772 - accuracy: 0.772
Epoch 8: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_36
3110/3110 [==============================] - 29s 9ms/step - loss: 0.6772 - accuracy:
Epoch 9/200
3107/3110 [============================>.] - ETA: 0s - loss: 0.6465 - accuracy: 0.783
Epoch 9: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_36
3110/3110 [==============================] - 28s 9ms/step - loss: 0.6464 - accuracy:
Epoch 10/200
3105/3110 [============================>.] - ETA: 0s - loss: 0.6177 - accuracy: 0.792
Epoch 10: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 30s 10ms/step - loss: 0.6180 - accuracy:
Epoch 11/200
3108/3110 [============================>.] - ETA: 0s - loss: 0.5966 - accuracy: 0.799
Epoch 11: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 29s 9ms/step - loss: 0.5967 - accuracy:
Epoch 12/200
3108/3110 [============================>.] - ETA: 0s - loss: 0.5775 - accuracy: 0.805
Epoch 12: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 28s 9ms/step - loss: 0.5776 - accuracy:
Epoch 13/200
3106/3110 [============================>.] - ETA: 0s - loss: 0.5556 - accuracy: 0.812
Epoch 13: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 28s 9ms/step - loss: 0.5557 - accuracy:
Epoch 14/200
3110/3110 [==============================] - ETA: 0s - loss: 0.5408 - accuracy: 0.818
Epoch 14: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 28s 9ms/step - loss: 0.5408 - accuracy:
Epoch 15/200
3105/3110 [============================>.] - ETA: 0s - loss: 0.5243 - accuracy: 0.823
Epoch 15: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 29s 9ms/step - loss: 0.5244 - accuracy:
Epoch 16/200
3106/3110 [============================>.] - ETA: 0s - loss: 0.5097 - accuracy: 0.829
Epoch 16: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 28s 9ms/step - loss: 0.5098 - accuracy:
Epoch 17/200
3105/3110 [============================>.] - ETA: 0s - loss: 0.4956 - accuracy: 0.832
Epoch 17: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 28s 9ms/step - loss: 0.4957 - accuracy:
Epoch 18/200
3108/3110 [============================>.] - ETA: 0s - loss: 0.4853 - accuracy: 0.837
Epoch 18: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 29s 9ms/step - loss: 0.4855 - accuracy:
Epoch 19/200
3105/3110 [============================>.] - ETA: 0s - loss: 0.4729 - accuracy: 0.840
Epoch 19: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 28s 9ms/step - loss: 0.4729 - accuracy:
Epoch 20/200
3104/3110 [============================>.] - ETA: 0s - loss: 0.4598 - accuracy: 0.844
Epoch 20: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 28s 9ms/step - loss: 0.4599 - accuracy:
Epoch 21/200
3104/3110 [============================>.] - ETA: 0s - loss: 0.4517 - accuracy: 0.847
Epoch 21: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 28s 9ms/step - loss: 0.4517 - accuracy:
Epoch 22/200
3109/3110 [============================>.] - ETA: 0s - loss: 0.4429 - accuracy: 0.850
Epoch 22: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 27s 9ms/step - loss: 0.4429 - accuracy:
Epoch 23/200
3108/3110 [============================>.] - ETA: 0s - loss: 0.4298 - accuracy: 0.854
```

```
Epoch 23: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 27s 9ms/step - loss: 0.4298 - accuracy:
Epoch 24/200
3105/3110 [============================>.] - ETA: 0s - loss: 0.4262 - accuracy: 0.855
Epoch 24: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 27s 9ms/step - loss: 0.4260 - accuracy:
Epoch 25/200
3105/3110 [============================>.] - ETA: 0s - loss: 0.4144 - accuracy: 0.858
Epoch 25: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 29s 9ms/step - loss: 0.4144 - accuracy:
Epoch 26/200
3106/3110 [============================>.] - ETA: 0s - loss: 0.4069 - accuracy: 0.860
Epoch 26: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 27s 9ms/step - loss: 0.4070 - accuracy:
Epoch 27/200
3108/3110 [============================>.] - ETA: 0s - loss: 0.4036 - accuracy: 0.864
Epoch 27: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 27s 9ms/step - loss: 0.4036 - accuracy:
Epoch 28/200
3108/3110 [============================>.] - ETA: 0s - loss: 0.3933 - accuracy: 0.866
Epoch 28: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 27s 9ms/step - loss: 0.3932 - accuracy:
Epoch 29/200
3107/3110 [============================>.] - ETA: 0s - loss: 0.3891 - accuracy: 0.868
Epoch 29: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 27s 9ms/step - loss: 0.3891 - accuracy:
Epoch 30/200
3105/3110 [============================>.] - ETA: 0s - loss: 0.3821 - accuracy: 0.868
Epoch 30: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 28s 9ms/step - loss: 0.3821 - accuracy:
Epoch 31/200
3110/3110 [==============================] - ETA: 0s - loss: 0.3756 - accuracy: 0.872
Epoch 31: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 28s 9ms/step - loss: 0.3756 - accuracy:
Epoch 32/200
3108/3110 [============================>.] - ETA: 0s - loss: 0.3729 - accuracy: 0.873
Epoch 32: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 28s 9ms/step - loss: 0.3730 - accuracy:
Epoch 33/200
3107/3110 [============================>.] - ETA: 0s - loss: 0.3630 - accuracy: 0.875
Epoch 33: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 28s 9ms/step - loss: 0.3629 - accuracy:
Epoch 34/200
3108/3110 [============================>.] - ETA: 0s - loss: 0.3582 - accuracy: 0.878
Epoch 34: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 29s 9ms/step - loss: 0.3582 - accuracy:
Epoch 35/200
3106/3110 [============================>.] - ETA: 0s - loss: 0.3534 - accuracy: 0.880
Epoch 35: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 28s 9ms/step - loss: 0.3535 - accuracy:
Epoch 36/200
3109/3110 [============================>.] - ETA: 0s - loss: 0.3512 - accuracy: 0.881
Epoch 36: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 28s 9ms/step - loss: 0.3511 - accuracy:
Epoch 37/200
3109/3110 [============================>.] - ETA: 0s - loss: 0.3442 - accuracy: 0.882
Epoch 37: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 28s 9ms/step - loss: 0.3443 - accuracy:
Epoch 38/200
3108/3110 [============================>.] - ETA: 0s - loss: 0.3411 - accuracy: 0.883
Epoch 38: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 28s 9ms/step - loss: 0.3411 - accuracy:
Epoch 39/200
3104/3110 [============================>.] - ETA: 0s - loss: 0.3382 - accuracy: 0.884
Epoch 39: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 29s 9ms/step - loss: 0.3382 - accuracy:
Epoch 40/200
3108/3110 [============================>.] - ETA: 0s - loss: 0.3317 - accuracy: 0.887
Epoch 40: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 28s 9ms/step - loss: 0.3317 - accuracy:
Epoch 41/200
3108/3110 [============================>.] - ETA: 0s - loss: 0.3265 - accuracy: 0.888
Epoch 41: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 30s 9ms/step - loss: 0.3266 - accuracy:
Epoch 42/200
3105/3110 [============================>.] - ETA: 0s - loss: 0.3238 - accuracy: 0.889
Epoch 42: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 31s 10ms/step - loss: 0.3239 - accuracy:
Epoch 43/200
3104/3110 [============================>.] - ETA: 0s - loss: 0.3202 - accuracy: 0.891
Epoch 43: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 29s 9ms/step - loss: 0.3203 - accuracy:
Epoch 44/200
3108/3110 [============================>.] - ETA: 0s - loss: 0.3109 - accuracy: 0.894
Epoch 44: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 29s 9ms/step - loss: 0.3110 - accuracy:
Epoch 45/200
3109/3110 [============================>.] - ETA: 0s - loss: 0.3122 - accuracy: 0.893
Epoch 45: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 30s 10ms/step - loss: 0.3122 - accuracy:
Epoch 46/200
```

```
Epoch 45/200
3110/3110 [==============================] - ETA: 0s - loss: 0.3054 - accuracy: 0.895
Epoch 46: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 33s 11ms/step - loss: 0.3054 - accuracy:
Epoch 47/200
3107/3110 [============================>.] - ETA: 0s - loss: 0.3043 - accuracy: 0.895
Epoch 47: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 33s 10ms/step - loss: 0.3044 - accuracy:
Epoch 48/200
3109/3110 [============================>.] - ETA: 0s - loss: 0.3023 - accuracy: 0.896
Epoch 48: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 33s 10ms/step - loss: 0.3023 - accuracy:
Epoch 49/200
3108/3110 [============================>.] - ETA: 0s - loss: 0.2980 - accuracy: 0.899
Epoch 49: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 32s 10ms/step - loss: 0.2980 - accuracy:
Epoch 50/200
3110/3110 [==============================] - ETA: 0s - loss: 0.2936 - accuracy: 0.900
Epoch 50: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 33s 11ms/step - loss: 0.2936 - accuracy:
Epoch 51/200
3107/3110 [============================>.] - ETA: 0s - loss: 0.2928 - accuracy: 0.900
Epoch 51: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 34s 11ms/step - loss: 0.2929 - accuracy:
Epoch 52/200
3106/3110 [============================>.] - ETA: 0s - loss: 0.2908 - accuracy: 0.901
Epoch 52: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 32s 10ms/step - loss: 0.2908 - accuracy:
Epoch 53/200
3110/3110 [==============================] - ETA: 0s - loss: 0.2845 - accuracy: 0.903
Epoch 53: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 33s 10ms/step - loss: 0.2845 - accuracy:
Epoch 54/200
3105/3110 [============================>.] - ETA: 0s - loss: 0.2834 - accuracy: 0.903
Epoch 54: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 29s 9ms/step - loss: 0.2835 - accuracy:
Epoch 55/200
3107/3110 [============================>.] - ETA: 0s - loss: 0.2826 - accuracy: 0.904
Epoch 55: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 31s 10ms/step - loss: 0.2827 - accuracy:
Epoch 56/200
3107/3110 [============================>.] - ETA: 0s - loss: 0.2775 - accuracy: 0.905
Epoch 56: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 33s 11ms/step - loss: 0.2774 - accuracy:
Epoch 57/200
3109/3110 [============================>.] - ETA: 0s - loss: 0.2785 - accuracy: 0.904
Epoch 57: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 32s 10ms/step - loss: 0.2786 - accuracy:
Epoch 58/200
3107/3110 [============================>.] - ETA: 0s - loss: 0.2755 - accuracy: 0.907
Epoch 58: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 32s 10ms/step - loss: 0.2756 - accuracy:
Epoch 59/200
3110/3110 [==============================] - ETA: 0s - loss: 0.2679 - accuracy: 0.909
Epoch 59: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 33s 10ms/step - loss: 0.2679 - accuracy:
Epoch 60/200
3108/3110 [============================>.] - ETA: 0s - loss: 0.2652 - accuracy: 0.909
Epoch 60: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 34s 11ms/step - loss: 0.2653 - accuracy:
Epoch 61/200
3106/3110 [============================>.] - ETA: 0s - loss: 0.2635 - accuracy: 0.910
Epoch 61: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 32s 10ms/step - loss: 0.2635 - accuracy:
Epoch 62/200
3108/3110 [============================>.] - ETA: 0s - loss: 0.2634 - accuracy: 0.910
Epoch 62: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 32s 10ms/step - loss: 0.2633 - accuracy:
Epoch 63/200
3110/3110 [==============================] - ETA: 0s - loss: 0.2598 - accuracy: 0.911
Epoch 63: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 33s 11ms/step - loss: 0.2598 - accuracy:
Epoch 64/200
3109/3110 [============================>.] - ETA: 0s - loss: 0.2557 - accuracy: 0.912
Epoch 64: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 32s 10ms/step - loss: 0.2557 - accuracy:
Epoch 65/200
3108/3110 [============================>.] - ETA: 0s - loss: 0.2565 - accuracy: 0.912
Epoch 65: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 31s 10ms/step - loss: 0.2564 - accuracy:
Epoch 66/200
3108/3110 [============================>.] - ETA: 0s - loss: 0.2550 - accuracy: 0.913
Epoch 66: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 32s 10ms/step - loss: 0.2550 - accuracy:
Epoch 67/200
3106/3110 [============================>.] - ETA: 0s - loss: 0.2510 - accuracy: 0.914
Epoch 67: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 33s 11ms/step - loss: 0.2510 - accuracy:
Epoch 68/200
3109/3110 [============================>.] - ETA: 0s - loss: 0.2518 - accuracy: 0.913
Epoch 68: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
```

```
3110/3110 [==============================] - 33s 11ms/step - loss: 0.2518 - accuracy:
Epoch 69/200
3107/3110 [============================>.] - ETA: 0s - loss: 0.2498 - accuracy: 0.913
Epoch 69: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 33s 10ms/step - loss: 0.2497 - accuracy:
Epoch 70/200
3108/3110 [============================>.] - ETA: 0s - loss: 0.2480 - accuracy: 0.915
Epoch 70: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 33s 11ms/step - loss: 0.2479 - accuracy:
Epoch 71/200
3110/3110 [==============================] - ETA: 0s - loss: 0.2445 - accuracy: 0.916
Epoch 71: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 33s 10ms/step - loss: 0.2445 - accuracy:
Epoch 72/200
3107/3110 [============================>.] - ETA: 0s - loss: 0.2409 - accuracy: 0.917
Epoch 72: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 31s 10ms/step - loss: 0.2410 - accuracy:
Epoch 73/200
3108/3110 [============================>.] - ETA: 0s - loss: 0.2406 - accuracy: 0.917
Epoch 73: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 32s 10ms/step - loss: 0.2406 - accuracy:
Epoch 74/200
3108/3110 [============================>.] - ETA: 0s - loss: 0.2378 - accuracy: 0.918
Epoch 74: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 31s 10ms/step - loss: 0.2378 - accuracy:
Epoch 75/200
3107/3110 [============================>.] - ETA: 0s - loss: 0.2385 - accuracy: 0.918
Epoch 75: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 34s 11ms/step - loss: 0.2386 - accuracy:
Epoch 76/200
3109/3110 [============================>.] - ETA: 0s - loss: 0.2301 - accuracy: 0.921
Epoch 76: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 32s 10ms/step - loss: 0.2300 - accuracy:
Epoch 77/200
3107/3110 [============================>.] - ETA: 0s - loss: 0.2335 - accuracy: 0.920
Epoch 77: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 32s 10ms/step - loss: 0.2335 - accuracy:
Epoch 78/200
3109/3110 [============================>.] - ETA: 0s - loss: 0.2362 - accuracy: 0.920
Epoch 78: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 32s 10ms/step - loss: 0.2362 - accuracy:
Epoch 79/200
3109/3110 [============================>.] - ETA: 0s - loss: 0.2308 - accuracy: 0.921
Epoch 79: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 33s 10ms/step - loss: 0.2308 - accuracy:
Epoch 80/200
3110/3110 [==============================] - ETA: 0s - loss: 0.2307 - accuracy: 0.921
Epoch 80: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 32s 10ms/step - loss: 0.2307 - accuracy:
Epoch 81/200
3107/3110 [============================>.] - ETA: 0s - loss: 0.2277 - accuracy: 0.923
Epoch 81: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 32s 10ms/step - loss: 0.2276 - accuracy:
Epoch 82/200
3109/3110 [============================>.] - ETA: 0s - loss: 0.2259 - accuracy: 0.923
Epoch 82: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 31s 10ms/step - loss: 0.2259 - accuracy:
Epoch 83/200
3106/3110 [============================>.] - ETA: 0s - loss: 0.2210 - accuracy: 0.924
Epoch 83: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 32s 10ms/step - loss: 0.2210 - accuracy:
Epoch 84/200
3110/3110 [==============================] - ETA: 0s - loss: 0.2267 - accuracy: 0.922
Epoch 84: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 32s 10ms/step - loss: 0.2267 - accuracy:
Epoch 85/200
3110/3110 [==============================] - ETA: 0s - loss: 0.2222 - accuracy: 0.924
Epoch 85: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 32s 10ms/step - loss: 0.2222 - accuracy:
Epoch 86/200
3110/3110 [==============================] - ETA: 0s - loss: 0.2207 - accuracy: 0.925
Epoch 86: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 32s 10ms/step - loss: 0.2207 - accuracy:
Epoch 87/200
3108/3110 [============================>.] - ETA: 0s - loss: 0.2183 - accuracy: 0.926
Epoch 87: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 32s 10ms/step - loss: 0.2183 - accuracy:
Epoch 88/200
3109/3110 [============================>.] - ETA: 0s - loss: 0.2187 - accuracy: 0.925
Epoch 88: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 33s 11ms/step - loss: 0.2188 - accuracy:
Epoch 89/200
3109/3110 [============================>.] - ETA: 0s - loss: 0.2175 - accuracy: 0.925
Epoch 89: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 31s 10ms/step - loss: 0.2175 - accuracy:
Epoch 90/200
3110/3110 [==============================] - ETA: 0s - loss: 0.2182 - accuracy: 0.926
Epoch 90: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 31s 10ms/step - loss: 0.2182 - accuracy:
Epoch 91/200
3106/3110 [============================>.] - ETA: 0s - loss: 0.2123 - accuracy: 0.928
```

```
Epoch 91: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 31s 10ms/step - loss: 0.2123 - accuracy:
Epoch 92/200
3108/3110 [============================>.] - ETA: 0s - loss: 0.2142 - accuracy: 0.927
Epoch 92: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 34s 11ms/step - loss: 0.2142 - accuracy:
Epoch 93/200
3105/3110 [============================>.] - ETA: 0s - loss: 0.2098 - accuracy: 0.929
Epoch 93: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 33s 10ms/step - loss: 0.2099 - accuracy:
Epoch 94/200
3108/3110 [============================>.] - ETA: 0s - loss: 0.2083 - accuracy: 0.929
Epoch 94: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 33s 11ms/step - loss: 0.2083 - accuracy:
Epoch 95/200
3109/3110 [============================>.] - ETA: 0s - loss: 0.2101 - accuracy: 0.929
Epoch 95: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 35s 11ms/step - loss: 0.2101 - accuracy:
Epoch 96/200
3107/3110 [============================>.] - ETA: 0s - loss: 0.2097 - accuracy: 0.929
Epoch 96: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 33s 10ms/step - loss: 0.2097 - accuracy:
Epoch 97/200
3108/3110 [============================>.] - ETA: 0s - loss: 0.2093 - accuracy: 0.929
Epoch 97: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 32s 10ms/step - loss: 0.2094 - accuracy:
Epoch 98/200
3109/3110 [============================>.] - ETA: 0s - loss: 0.2084 - accuracy: 0.929
Epoch 98: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 32s 10ms/step - loss: 0.2084 - accuracy:
Epoch 99/200
3105/3110 [============================>.] - ETA: 0s - loss: 0.2045 - accuracy: 0.930
Epoch 99: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_3
3110/3110 [==============================] - 32s 10ms/step - loss: 0.2045 - accuracy:
Epoch 100/200
3106/3110 [============================>.] - ETA: 0s - loss: 0.2053 - accuracy: 0.930
Epoch 100: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_
3110/3110 [==============================] - 33s 10ms/step - loss: 0.2054 - accuracy:
Epoch 101/200
3110/3110 [==============================] - ETA: 0s - loss: 0.2057 - accuracy: 0.931
Epoch 101: saving model to drive/MyDrive/Colab Notebooks/Cifar10/saves/18_07_2023-18_
3110/3110 [==============================] - 35s 11ms/step - loss: 0.2057 - accuracy:
Epoch 102/200
 615/3110 [====>.........................] - ETA: 23s - loss: 0.2032 - accuracy: 0.93
---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
<ipython-input-17-1903f564c0b1> in <cell line: 1>()
----> 1 model.fit(newdata,epochs=200,callbacks=[sav])

                                 11 frames
/usr/local/lib/python3.10/dist-packages/tensorflow/python/util/nest.py in
flatten(structure, expand_composites)
    354
    355
--> 356 @tf_export("nest.flatten")
    357 def flatten(structure, expand_composites=False):
    358   """Returns a flat list from a given structure.
```

```
a,b=to_tensors(x_test,y_test)
model.evaluate(a,b)
##accuracy is 86.33% in test data
```

```
313/313 [==============================] - 1s 4ms/step - loss: 0.4320 - accuracy: 0.8634
[0.43199530243873596, 0.8633999824523926]
```

```
model.save("drive/MyDrive/Colab Notebooks/Cifar10/saves/without_valid",save_format='tf')
```

```
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_
```

```
model=tf.keras.models.load_model("drive/MyDrive/Colab Notebooks/Cifar10/saves/without_valid")
```

```
# will run for 20 more epochs and see if there's any improvement
model.fit(newdata,epochs=20)
```

```
Epoch 1/20
3110/3110 [==============================] - 40s 8ms/step - loss: 0.3232 - accuracy: 0.8953
Epoch 2/20
3110/3110 [==============================] - 27s 9ms/step - loss: 0.3081 - accuracy: 0.8992
Epoch 3/20
3110/3110 [==============================] - 26s 8ms/step - loss: 0.2986 - accuracy: 0.9023
```

```
Epoch 4/20
3110/3110 [==============================] - 27s 9ms/step - loss: 0.2921 - accuracy: 0.9041
Epoch 5/20
3110/3110 [==============================] - 25s 8ms/step - loss: 0.2879 - accuracy: 0.9056
Epoch 6/20
3110/3110 [==============================] - 25s 8ms/step - loss: 0.2815 - accuracy: 0.9067
Epoch 7/20
3110/3110 [==============================] - 25s 8ms/step - loss: 0.2749 - accuracy: 0.9086
Epoch 8/20
3110/3110 [==============================] - 26s 8ms/step - loss: 0.2719 - accuracy: 0.9100
Epoch 9/20
3110/3110 [==============================] - 26s 8ms/step - loss: 0.2657 - accuracy: 0.9110
Epoch 10/20
3110/3110 [==============================] - 25s 8ms/step - loss: 0.2644 - accuracy: 0.9108
Epoch 11/20
3110/3110 [==============================] - 25s 8ms/step - loss: 0.2614 - accuracy: 0.9121
Epoch 12/20
3110/3110 [==============================] - 25s 8ms/step - loss: 0.2552 - accuracy: 0.9157
Epoch 13/20
3110/3110 [==============================] - 26s 8ms/step - loss: 0.2571 - accuracy: 0.9148
Epoch 14/20
3110/3110 [==============================] - 26s 8ms/step - loss: 0.2481 - accuracy: 0.9174
Epoch 15/20
3110/3110 [==============================] - 25s 8ms/step - loss: 0.2473 - accuracy: 0.9161
Epoch 16/20
3110/3110 [==============================] - 25s 8ms/step - loss: 0.2414 - accuracy: 0.9201
Epoch 17/20
3110/3110 [==============================] - 25s 8ms/step - loss: 0.2454 - accuracy: 0.9191
Epoch 18/20
3110/3110 [==============================] - 26s 8ms/step - loss: 0.2399 - accuracy: 0.9198
Epoch 19/20
3110/3110 [==============================] - 25s 8ms/step - loss: 0.2304 - accuracy: 0.9230
Epoch 20/20
3110/3110 [==============================] - 25s 8ms/step - loss: 0.2330 - accuracy: 0.9217
<keras.callbacks.History at 0x7b0ac3057d60>
```

```
a,b=to_tensors(x_test,y_test)
model.evaluate(a,b)
##the accuracy after 20 more epochs is almost same
```

```
313/313 [==============================] - 2s 5ms/step - loss: 0.4227 - accuracy: 0.8616
[0.42270898818969727, 0.8615999817848206]
```

```
a,b=to_tensors(x_test,y_test)
y_preds=model.predict(a)
y_preds=[np.argmax(x) for x in y_preds]
y_true=[np.argmax(x) for x in y_test]
```

```
313/313 [==============================] - 8s 4ms/step
```

```
print(classification_report(y_preds,y_true))
```

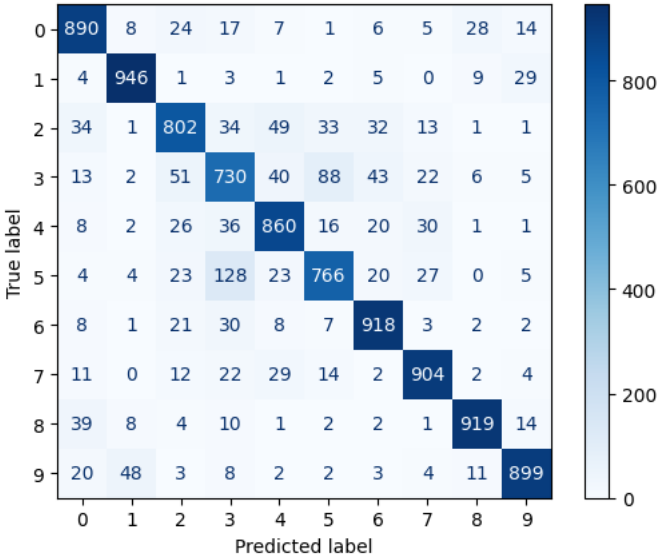
```
              precision    recall  f1-score   support

           0       0.89      0.86      0.88      1031
           1       0.95      0.93      0.94      1020
           2       0.80      0.83      0.82       967
           3       0.73      0.72      0.72      1018
           4       0.86      0.84      0.85      1020
           5       0.77      0.82      0.79       931
           6       0.92      0.87      0.90      1051
           7       0.90      0.90      0.90      1009
           8       0.92      0.94      0.93       979
           9       0.90      0.92      0.91       974

    accuracy                           0.86     10000
   macro avg       0.86      0.86      0.86     10000
weighted avg       0.86      0.86      0.86     10000
```

```
print(confusion_matrix(y_preds,y_true))
```

```
[[890   4  34  13   8   4   8  11  39  20]
 [  8 946   1   2   2   4   1   0   8  48]
 [ 24   1 802  51  26  23  21  12   4   3]
 [ 17   3  34 730  36 128  30  22  10   8]
 [  7   1  49  40 860  23   8  29   1   2]
 [  1   2  33  88  16 766   7  14   2   2]
 [  6   5  32  43  20  20 918   2   2   3]
 [  5   0  13  22  30  27   3 904   1   4]
 [ 28   9   1   6   1   0   2   2 919  11]
 [ 14  29   1   5   1   5   2   4  14 899]]
```

```
ConfusionMatrixDisplay.from_predictions(y_true=y_true,y_pred=y_preds,cmap='Blues')
```

```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7bc2fb148130>
```



Accuracy in test data in with no validation set is 86.33%

Testing with random Google image:

```
from PIL import Image


classes = ['airplane' ,'automobile', 'bird' , 'cat' , 'deer' ,'dog' ,'frog', 'horse' ,'ship' ,'truck']


image=Image.open("drive/MyDrive/Colab Notebooks/Cifar10/1.jpg")
image.show()
image = image.resize((32,32), Image.ANTIALIAS)

a=tf.convert_to_tensor(image)
a=a/255
a=tf.expand_dims(a,axis=0)
prediction=model.predict(a)
classes[prediction.argmax()]
```

```
1/1 [==============================] - 0s 44ms/step
'dog'
```

```
image=Image.open("drive/MyDrive/Colab Notebooks/Cifar10/2.webp")
image.show()
image = image.resize((32,32), Image.ANTIALIAS)

a=tf.convert_to_tensor(image)
a=a/255
a=tf.expand_dims(a,axis=0)
prediction=model.predict(a)
classes[prediction.argmax()]
```

```
1/1 [==============================] - 0s 34ms/step
'ship'
```

```
image=Image.open("drive/MyDrive/Colab Notebooks/Cifar10/3.jpg")
image.show()
image = image.resize((32,32), Image.ANTIALIAS)

a=tf.convert_to_tensor(image)
a=a/255
a=tf.expand_dims(a,axis=0)
prediction=model.predict(a)
print(classes[prediction.argmax()])
```

```
1/1 [==============================] - 0s 45ms/step
automobile
```

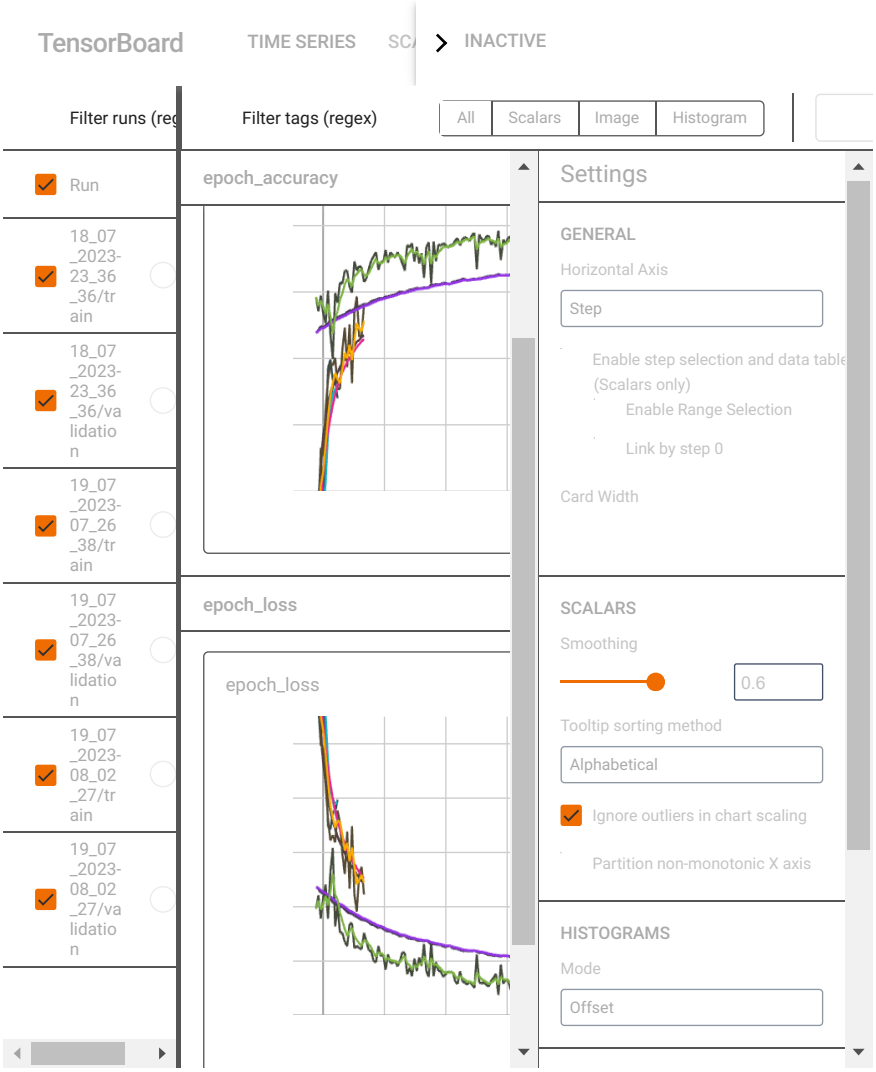
```
image=Image.open("drive/MyDrive/Colab Notebooks/Cifar10/4.jpg")
image.show()
image = image.resize((32,32), Image.ANTIALIAS)

a=tf.convert_to_tensor(image)
a=a/255
a=tf.expand_dims(a,axis=0)
prediction=model.predict(a)
classes[prediction.argmax()]
```

```
1/1 [==============================] - 0s 30ms/step
'truck'
```

4 Tensorboard visualisation for training with validation set For CNN Model

```
%load_ext tensorboard
```

```
%tensorboard --logdir drive/MyDrive/Colab\ Notebooks/Cifar10/logtb
```



5 **Custom Residual Model**

```
import tensorflow as tf
from tensorflow.keras.layers import Input, Conv2D, BatchNormalization, Activation, Add, GlobalAveragePooling2D, Dense
from tensorflow.keras.models import Model

def residual_block(x, filters, downsample=False):
    # Shortcut branch
    shortcut = x

    # First conv layer
    strides = (2, 2) if downsample else (1, 1)
    x = Conv2D(filters, kernel_size=(3, 3), strides=strides, padding='same')(x)
    x = BatchNormalization()(x)
    x = Activation('relu')(x)

    # Second conv layer
    x = Conv2D(filters, kernel_size=(3, 3), padding='same')(x)
    x = BatchNormalization()(x)

    # Downsample the shortcut to match the shape of x
    if downsample:
        shortcut = Conv2D(filters, kernel_size=(1, 1), strides=(2, 2), padding='same')(shortcut)
        shortcut = BatchNormalization()(shortcut)

    # Add the shortcut to the main path
    x = Add()([x, shortcut])
    x = Activation('relu')(x)
    return x



def create_resnet_model(input_shape, num_classes):
    inputs = Input(shape=input_shape)

    # Initial conv layer
    x = Conv2D(64, kernel_size=(3, 3), strides=(1, 1), padding='same')(inputs)
    x = BatchNormalization()(x)
    x = Activation('relu')(x)

    # Residual blocks
    x = residual_block(x, filters=64)
    x = residual_block(x, filters=64)
    x = residual_block(x, filters=128, downsample=True)
    x = residual_block(x, filters=128)
    x = residual_block(x, filters=256, downsample=True)
    x = residual_block(x, filters=256)

    # Global average pooling and final dense layer
    x = GlobalAveragePooling2D()(x)
    outputs = Dense(num_classes, activation='softmax')(x)

    model = Model(inputs=inputs, outputs=outputs)
    return model

# Create the ResNet model for CIFAR-10
input_shape = (32, 32, 3)
num_classes = 10
resnet_model = create_resnet_model(input_shape, num_classes)

# Compile the model
resnet_model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
resnet_model.summary()
```

```
Model: "model"
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_2 (InputLayer)           [(None, 32, 32, 3)]  0           []

 conv2d (Conv2D)                (None, 32, 32, 64)   1792        ['input_2[0][0]']

 batch_normalization (BatchNorm  (None, 32, 32, 64)  256         ['conv2d[0][0]']
 alization)

 activation (Activation)        (None, 32, 32, 64)   0           ['batch_normalization[0][0]']

 conv2d_1 (Conv2D)              (None, 32, 32, 64)   36928       ['activation[0][0]']

 batch_normalization_1 (BatchNo  (None, 32, 32, 64)  256         ['conv2d_1[0][0]']
 rmalization)

 activation_1 (Activation)      (None, 32, 32, 64)   0           ['batch_normalization_1[0][0]']

 conv2d_2 (Conv2D)              (None, 32, 32, 64)   36928       ['activation_1[0][0]']
```

```
batch_normalization_2 (BatchNo  (None, 32, 32, 64)  256         ['conv2d_2[0][0]']
rmalization)

add (Add)                       (None, 32, 32, 64)   0          ['batch_normalization_2[0][0]',
                                                                 'activation[0][0]']

activation_2 (Activation)       (None, 32, 32, 64)   0          ['add[0][0]']

conv2d_3 (Conv2D)               (None, 32, 32, 64)   36928      ['activation_2[0][0]']

batch_normalization_3 (BatchNo  (None, 32, 32, 64)  256         ['conv2d_3[0][0]']
rmalization)

activation_3 (Activation)       (None, 32, 32, 64)   0          ['batch_normalization_3[0][0]']

conv2d_4 (Conv2D)               (None, 32, 32, 64)   36928      ['activation_3[0][0]']

batch_normalization_4 (BatchNo  (None, 32, 32, 64)  256         ['conv2d_4[0][0]']
rmalization)

add_1 (Add)                     (None, 32, 32, 64)   0          ['batch_normalization_4[0][0]',
                                                                 'activation_2[0][0]']

activation_4 (Activation)       (None, 32, 32, 64)   0          ['add_1[0][0]']

conv2d_5 (Conv2D)               (None, 16, 16, 128)  73856      ['activation_4[0][0]']

batch_normalization_5 (BatchNo  (None, 16, 16, 128)  512        ['conv2d_5[0][0]']
rmalization)

activation_5 (Activation)       (None, 16, 16, 128)  0          ['batch_normalization_5[0][0]']

conv2d_6 (Conv2D)               (None, 16, 16, 128)  147584     ['activation_5[0][0]']

conv2d_7 (Conv2D)               (None, 16, 16, 128)  8320       ['activation_4[0][0]']
```

```
data=preprocessing(x_train,y_train)
```

```
history=resnet_model.fit(data,epochs=30)
```

```
Epoch 2/30
1563/1563 [==============================] - 44s 28ms/step - loss: 0.6464 - accuracy: 0.7731
Epoch 3/30
1563/1563 [==============================] - 44s 28ms/step - loss: 0.5182 - accuracy: 0.8206
Epoch 4/30
1563/1563 [==============================] - 44s 28ms/step - loss: 0.4285 - accuracy: 0.8515
Epoch 5/30
1563/1563 [==============================] - 44s 28ms/step - loss: 0.3426 - accuracy: 0.8802
Epoch 6/30
1563/1563 [==============================] - 44s 28ms/step - loss: 0.2758 - accuracy: 0.9044
Epoch 7/30
1563/1563 [==============================] - 44s 28ms/step - loss: 0.2151 - accuracy: 0.9243
Epoch 8/30
1563/1563 [==============================] - 44s 28ms/step - loss: 0.1604 - accuracy: 0.9434
Epoch 9/30
1563/1563 [==============================] - 44s 28ms/step - loss: 0.1282 - accuracy: 0.9549
Epoch 10/30
1563/1563 [==============================] - 44s 28ms/step - loss: 0.1014 - accuracy: 0.9648
Epoch 11/30
1563/1563 [==============================] - 44s 28ms/step - loss: 0.0855 - accuracy: 0.9705
Epoch 12/30
1563/1563 [==============================] - 44s 28ms/step - loss: 0.0764 - accuracy: 0.9731
Epoch 13/30
1563/1563 [==============================] - 44s 28ms/step - loss: 0.0649 - accuracy: 0.9778
Epoch 14/30
1563/1563 [==============================] - 44s 28ms/step - loss: 0.0635 - accuracy: 0.9780
Epoch 15/30
1563/1563 [==============================] - 44s 28ms/step - loss: 0.0549 - accuracy: 0.9804
Epoch 16/30
1563/1563 [==============================] - 45s 29ms/step - loss: 0.0482 - accuracy: 0.9828
Epoch 17/30
1563/1563 [==============================] - 44s 28ms/step - loss: 0.0464 - accuracy: 0.9833
Epoch 18/30
1563/1563 [==============================] - 44s 28ms/step - loss: 0.0432 - accuracy: 0.9844
Epoch 19/30
1563/1563 [==============================] - 44s 28ms/step - loss: 0.0423 - accuracy: 0.9854
Epoch 20/30
1563/1563 [==============================] - 44s 28ms/step - loss: 0.0390 - accuracy: 0.9869
Epoch 21/30
1563/1563 [==============================] - 44s 28ms/step - loss: 0.0369 - accuracy: 0.9873
Epoch 22/30
1563/1563 [==============================] - 44s 28ms/step - loss: 0.0305 - accuracy: 0.9892
Epoch 23/30
1563/1563 [==============================] - 44s 28ms/step - loss: 0.0343 - accuracy: 0.9887
Epoch 24/30
```

```
1563/1563 [==============================] - 45s 29ms/step - loss: 0.0287 - accuracy: 0.9902
Epoch 26/30
1563/1563 [==============================] - 44s 28ms/step - loss: 0.0288 - accuracy: 0.9904
Epoch 27/30
1563/1563 [==============================] - 44s 28ms/step - loss: 0.0271 - accuracy: 0.9905
Epoch 28/30
1563/1563 [==============================] - 45s 28ms/step - loss: 0.0270 - accuracy: 0.9911
Epoch 29/30
1563/1563 [==============================] - 44s 28ms/step - loss: 0.0262 - accuracy: 0.9910
Epoch 30/30
1563/1563 [==============================] - 45s 29ms/step - loss: 0.0279 - accuracy: 0.9910
```

```
a,b=to_tensors(x_test,y_test)
resnet_model.evaluate(a,b)
```

```
313/313 [==============================] - 4s 11ms/step - loss: 0.8147 - accuracy: 0.8408
[0.8146928548812866, 0.8407999873161316]
```

```
print(classification_report(y_preds,y_true))
```

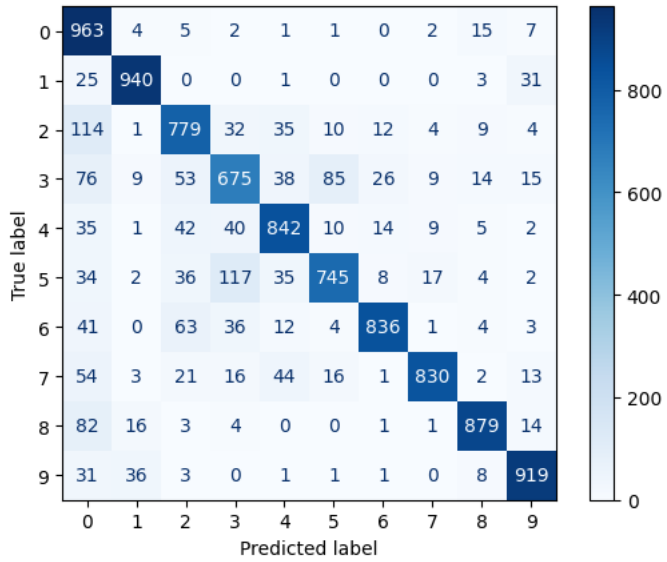
```
              precision    recall  f1-score   support

           0       0.96      0.66      0.78      1455
           1       0.94      0.93      0.93      1012
           2       0.78      0.78      0.78      1005
           3       0.68      0.73      0.70       922
           4       0.84      0.83      0.84      1009
           5       0.74      0.85      0.80       872
           6       0.84      0.93      0.88       899
           7       0.83      0.95      0.89       873
           8       0.88      0.93      0.90       943
           9       0.92      0.91      0.91      1010

    accuracy                           0.84     10000
   macro avg       0.84      0.85      0.84     10000
weighted avg       0.85      0.84      0.84     10000
```

```
print(confusion_matrix(y_preds,y_true))
```

```
[[963  25 114  76  35  34  41  54  82  31]
 [  4 940   1   9   1   2   0   3  16  36]
 [  5   0 779  53  42  36  63  21   3   3]
 [  2   0  32 675  40 117  36  16   4   0]
 [  1   1  35  38 842  35  12  44   0   1]
 [  1   0  10  85  10 745   4  16   0   1]
 [  0   0  12  26  14   8 836   1   1   1]
 [  2   0   4   9   9  17   1 830   1   0]
 [ 15   3   9  14   5   4   4   2 879   8]
 [  7  31   4  15   2   2   3  13  14 919]]
```

```
ConfusionMatrixDisplay.from_predictions(y_true=y_true,y_pred=y_preds,cmap='Blues')
```

```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7acbc034e5f0>
```




```
resnet_model.save("drive/MyDrive/Colab Notebooks/Cifar10/saves/resnet_model",save_format='tf')
```

```
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_
```

```
a,b=to_tensors(x_test,y_test)
```