

```
In [ ]: # import some libraries
import numpy as np
import tensorflow as tf
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
```

```
In [ ]: # Load the dataset
num_classes = 10
cifar10 = tf.keras.datasets.cifar10
(x_learn, y_learn), (x_test, y_test) = cifar10.load_data()

# Normalize the data in [0 1]
print("Normalizing training set..")
x_learn = np.asarray(x_learn, dtype=np.float32) / 255 # Normalizing training set
print("Normalizing test set..")
x_test = np.asarray(x_test, dtype=np.float32) / 255 # Normalizing test set
```

Normalizing training set..  
Normalizing test set..

```
In [ ]: # split in training and validation
x_train, x_val, y_train, y_val = train_test_split(x_learn, y_learn, test_size=0.25)
```

```
In [ ]: # Standardizing the data
def standardize_dataset(X):
    image_means = []
    image_stds = []

    for image in X:
        image_means.append(np.mean(image)) # Computing the image mean
        image_stds.append(np.std(image)) # Computing the image standard deviation

    dataset_mean = np.mean(image_means) # Computing the dataset mean
    dataset_std = np.mean(image_stds) # Computing the dataset standard deviation
    return [dataset_mean, dataset_std] # For every image we subtract to it the dataset mean and divide by the dataset std
```

dataset\_mean, dataset\_std = standardize\_dataset(x\_train)

```
print("Standardizing training set..")
x_train = (x_train-dataset_mean)/dataset_std # Standardizing the training set
print("Standardizing validation set..")
x_val = (x_val-dataset_mean)/dataset_std # Standardizing the test set
print("Standardizing test set..")
x_test = (x_test-dataset_mean)/dataset_std # Standardizing the test set
```

```
# one hot encode target values
y_train = tf.keras.utils.to_categorical(y_train)
y_val = tf.keras.utils.to_categorical(y_val)
y_test = tf.keras.utils.to_categorical(y_test)
```

```
print("Size of the training set")
print("x_train", x_train.shape)
print("y_train", y_train.shape)

print("Size of the validation set")
print("x_val", x_val.shape)
print("y_val", y_val.shape)

print("Size of the test set")
print("x_test", x_test.shape)
print("y_test", y_test.shape)
```

```
Standardizing training set..  
Standardizing validation set..  
Standardizing test set..  
Size of the training set  
x_train (37500, 32, 32, 3)  
y_train (37500, 10)  
Size of the validation set  
x_val (12500, 32, 32, 3)  
y_val (12500, 10)  
Size of the test set  
x_test (10000, 32, 32, 3)  
y_test (10000, 10)
```

## Plot some sample from the training set

```
In [ ]: class_names = ['Airplane', 'Automobile', 'Bird', 'Cat', 'Deer',  
                  'Dog', 'Frog', 'Horse', 'Ship', 'Truck']  
  
plt.figure(figsize=(6,6))  
for i in range(16):  
    plt.subplot(4,4,i+1)  
    plt.xticks([])  
    plt.yticks([])  
    plt.grid(False)  
    plt.imshow(x_learn[i], cmap=plt.cm.binary)  
    plt.xlabel(class_names[y_learn[i,0]])  
plt.show()
```



## Build a Model

```
In [ ]: from tensorflow import keras  
from keras.layers import Flatten  
from keras.models import Sequential  
from keras.layers import Dense  
from keras import layers, regularizers  
from keras.layers import Dropout, Conv2D, Activation, MaxPooling2D, BatchNormalization  
  
tf.keras.backend.clear_session()
```

```
np.random.seed(42)
tf.random.set_seed(42)

num_filters_1=32
num_filters_2=64
num_filters_3=128
filter_size=3
pool_size=2

l2_norm = .0001
model = Sequential()
model.add(Conv2D(num_filters_1,filter_size, padding='same',
                kernel_regularizer=regularizers.l2(l2_norm), input_shape=x_train.shape))
model.add(BatchNormalization())

model.add(Conv2D(num_filters_1, filter_size, padding='same',
                kernel_regularizer=regularizers.l2(l2_norm),activation='elu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=pool_size))
model.add(Dropout(0.2))
model.add(Conv2D(num_filters_2, filter_size, padding='same',
                kernel_regularizer=regularizers.l2(l2_norm),activation='elu'))
model.add(BatchNormalization())

model.add(Conv2D(num_filters_2, filter_size, padding='same',
                kernel_regularizer=regularizers.l2(l2_norm),activation='elu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=pool_size))
model.add(Dropout(0.3))
model.add(Conv2D(num_filters_3, filter_size, padding='same',
                kernel_regularizer=regularizers.l2(l2_norm),activation='elu'))
model.add(BatchNormalization())
model.add(Conv2D(num_filters_3, filter_size, padding='same',
                kernel_regularizer=regularizers.l2(l2_norm),activation='elu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.4))

model.add(Flatten())
model.add(Dense(num_classes, activation='softmax'))

model.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
<hr/>		
conv2d (Conv2D)	(None, 32, 32, 32)	896
batch_normalization (BatchNormalization)	(None, 32, 32, 32)	128
conv2d_1 (Conv2D)	(None, 32, 32, 32)	9248
<hr/>		
Layer (type)	Output Shape	Param #
<hr/>		
conv2d (Conv2D)	(None, 32, 32, 32)	896
batch_normalization (BatchNormalization)	(None, 32, 32, 32)	128
conv2d_1 (Conv2D)	(None, 32, 32, 32)	9248
batch_normalization_1 (BatchNormalization)	(None, 32, 32, 32)	128
max_pooling2d (MaxPooling2D)	(None, 16, 16, 32)	0
dropout (Dropout)	(None, 16, 16, 32)	0
conv2d_2 (Conv2D)	(None, 16, 16, 64)	18496
batch_normalization_2 (BatchNormalization)	(None, 16, 16, 64)	256
conv2d_3 (Conv2D)	(None, 16, 16, 64)	36928
batch_normalization_3 (BatchNormalization)	(None, 16, 16, 64)	256
max_pooling2d_1 (MaxPooling2D)	(None, 8, 8, 64)	0
dropout_1 (Dropout)	(None, 8, 8, 64)	0
conv2d_4 (Conv2D)	(None, 8, 8, 128)	73856
batch_normalization_4 (BatchNormalization)	(None, 8, 8, 128)	512
conv2d_5 (Conv2D)	(None, 8, 8, 128)	147584
batch_normalization_5 (BatchNormalization)	(None, 8, 8, 128)	512
max_pooling2d_2 (MaxPooling2D)	(None, 4, 4, 128)	0
dropout_2 (Dropout)	(None, 4, 4, 128)	0
flatten (Flatten)	(None, 2048)	0
dense (Dense)	(None, 10)	20490
<hr/>		

```
Total params: 309,290  
Trainable params: 308,394  
Non-trainable params: 896
```

```
In [ ]: from keras.optimizers import Adam, SGD, Adadelta, Adagrad, Adamax, Nadam, RMSprop  
  
rms = RMSprop(learning_rate=0.001,rho=0.9,epsilon=None,decay=0.000001)  
# Losses https://keras.io/Losses/  
loss = ['categorical_crossentropy']  
  
# Metrics https://www.tensorflow.org/api_docs/python/tf/metrics  
metrics = ['accuracy','precision','recall']  
  
# Compile the model you created before using  
# rms optimizer as optimizer  
# categorical crossentropy as Loss function  
# accuracy as metric  
def lr_schedule(epoch):  
    lrate = 0.001  
    if epoch > 30:  
        lrate = 0.0005  
    if epoch > 50:  
        lrate = 0.0003  
    return lrate  
  
model.compile(optimizer=rms,  
              loss=loss[0],  
              metrics=[metrics[0]],  
              )
```

```
In [ ]: batch_size = 64  
epochs = 70  
from tensorflow import keras  
from keras.callbacks import ModelCheckpoint, EarlyStopping  
import os  
  
from keras.preprocessing.image import ImageDataGenerator  
datagen = ImageDataGenerator( rotation_range=90,  
                             width_shift_range=0.1, height_shift_range=0.1,  
                             horizontal_flip=True)  
datagen.fit(x_train)  
  
es_callback = EarlyStopping(monitor='val_accuracy', patience=10, verbose=1)  
  
checkpoint_path = "output/cp.ckpt"  
checkpoint_dir = os.path.dirname(checkpoint_path)  
cp_callback = ModelCheckpoint(checkpoint_path, monitor='val_accuracy', save_best_only=True)  
  
history = model.fit(datagen.flow(x_train,y_train,batch_size=batch_size),  
                     validation_data=(x_val, y_val), epochs=epochs,  
                     callbacks=[keras.callbacks.LearningRateScheduler(lr_schedule),es_callback])
```

```
Epoch 1/70  
586/586 [=====] - ETA: 0s - loss: 2.2904 - accuracy: 0.31  
82
```

```
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op while saving (showing 5 of 6). These functions will not be directly callable after loading.
```

```
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
```

```
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
```

```
586/586 [=====] - 149s 251ms/step - loss: 2.2904 - accuracy: 0.3182 - val_loss: 1.8949 - val_accuracy: 0.4348 - lr: 0.0010
Epoch 2/70
586/586 [=====] - ETA: 0s - loss: 1.7793 - accuracy: 0.4304
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op while saving (showing 5 of 6). These functions will not be directly callable after loading.
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
586/586 [=====] - 193s 330ms/step - loss: 1.7793 - accuracy: 0.4304 - val_loss: 1.6819 - val_accuracy: 0.4770 - lr: 0.0010
Epoch 3/70
586/586 [=====] - ETA: 0s - loss: 1.5910 - accuracy: 0.4864
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op while saving (showing 5 of 6). These functions will not be directly callable after loading.
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
586/586 [=====] - 156s 266ms/step - loss: 1.5910 - accuracy: 0.4864 - val_loss: 1.5859 - val_accuracy: 0.5290 - lr: 0.0010
Epoch 4/70
586/586 [=====] - ETA: 0s - loss: 1.4624 - accuracy: 0.5230
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op while saving (showing 5 of 6). These functions will not be directly callable after loading.
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
586/586 [=====] - 195s 334ms/step - loss: 1.4624 - accuracy: 0.5230 - val_loss: 1.4711 - val_accuracy: 0.5489 - lr: 0.0010
Epoch 5/70
586/586 [=====] - ETA: 0s - loss: 1.3741 - accuracy: 0.5525
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op while saving (showing 5 of 6). These functions will not be directly callable after loading.
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
586/586 [=====] - 194s 332ms/step - loss: 1.3741 - accuracy: 0.5525 - val_loss: 1.5105 - val_accuracy: 0.5538 - lr: 0.0010
Epoch 6/70
586/586 [=====] - ETA: 0s - loss: 1.3045 - accuracy: 0.5754
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op while saving (showing 5 of 6). These functions will not be directly callable after loading.
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
```

```
586/586 [=====] - 190s 323ms/step - loss: 1.3045 - accuracy: 0.5754 - val_loss: 1.1847 - val_accuracy: 0.6243 - lr: 0.0010
Epoch 7/70
586/586 [=====] - 190s 324ms/step - loss: 1.2589 - accuracy: 0.5923 - val_loss: 1.5017 - val_accuracy: 0.5594 - lr: 0.0010
Epoch 8/70
586/586 [=====] - 189s 322ms/step - loss: 1.2093 - accuracy: 0.6096 - val_loss: 1.2749 - val_accuracy: 0.6094 - lr: 0.0010
Epoch 9/70
586/586 [=====] - ETA: 0s - loss: 1.1763 - accuracy: 0.6241
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op while saving (showing 5 of 6). These functions will not be directly callable after loading.
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
586/586 [=====] - 189s 323ms/step - loss: 1.1763 - accuracy: 0.6241 - val_loss: 1.1872 - val_accuracy: 0.6406 - lr: 0.0010
Epoch 10/70
586/586 [=====] - 176s 301ms/step - loss: 1.1505 - accuracy: 0.6325 - val_loss: 1.2134 - val_accuracy: 0.6323 - lr: 0.0010
Epoch 11/70
586/586 [=====] - ETA: 0s - loss: 1.1324 - accuracy: 0.6383
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op while saving (showing 5 of 6). These functions will not be directly callable after loading.
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
586/586 [=====] - 177s 301ms/step - loss: 1.1324 - accuracy: 0.6383 - val_loss: 1.1780 - val_accuracy: 0.6492 - lr: 0.0010
Epoch 12/70
586/586 [=====] - 184s 313ms/step - loss: 1.1116 - accuracy: 0.6501 - val_loss: 1.2247 - val_accuracy: 0.6412 - lr: 0.0010
Epoch 13/70
586/586 [=====] - 192s 327ms/step - loss: 1.0938 - accuracy: 0.6567 - val_loss: 1.3006 - val_accuracy: 0.6135 - lr: 0.0010
Epoch 14/70
586/586 [=====] - ETA: 0s - loss: 1.0838 - accuracy: 0.6603
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op while saving (showing 5 of 6). These functions will not be directly callable after loading.
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
586/586 [=====] - 178s 303ms/step - loss: 1.0838 - accuracy: 0.6603 - val_loss: 1.1066 - val_accuracy: 0.6738 - lr: 0.0010
Epoch 15/70
586/586 [=====] - 165s 281ms/step - loss: 1.0676 - accuracy: 0.6666 - val_loss: 1.1308 - val_accuracy: 0.6712 - lr: 0.0010
Epoch 16/70
586/586 [=====] - 171s 291ms/step - loss: 1.0602 - accuracy: 0.6714 - val_loss: 1.2396 - val_accuracy: 0.6434 - lr: 0.0010
Epoch 17/70
586/586 [=====] - 166s 283ms/step - loss: 1.0455 - accuracy: 0.6776 - val_loss: 1.1288 - val_accuracy: 0.6662 - lr: 0.0010
Epoch 18/70
586/586 [=====] - ETA: 0s - loss: 1.0342 - accuracy: 0.6848
```

```
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op while saving (showing 5 of 6). These functions will not be directly callable after loading.
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
586/586 [=====] - 167s 284ms/step - loss: 1.0342 - accuracy: 0.6848 - val_loss: 0.9912 - val_accuracy: 0.7126 - lr: 0.0010
Epoch 19/70
586/586 [=====] - 163s 278ms/step - loss: 1.0224 - accuracy: 0.6869 - val_loss: 1.1117 - val_accuracy: 0.6769 - lr: 0.0010
Epoch 20/70
586/586 [=====] - 163s 279ms/step - loss: 1.0157 - accuracy: 0.6899 - val_loss: 1.0107 - val_accuracy: 0.7083 - lr: 0.0010
Epoch 21/70
586/586 [=====] - ETA: 0s - loss: 1.0148 - accuracy: 0.6910
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op while saving (showing 5 of 6). These functions will not be directly callable after loading.
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
586/586 [=====] - 172s 293ms/step - loss: 1.0148 - accuracy: 0.6910 - val_loss: 0.9772 - val_accuracy: 0.7189 - lr: 0.0010
Epoch 22/70
586/586 [=====] - 163s 279ms/step - loss: 1.0015 - accuracy: 0.6965 - val_loss: 1.2353 - val_accuracy: 0.6581 - lr: 0.0010
Epoch 23/70
586/586 [=====] - ETA: 0s - loss: 0.9961 - accuracy: 0.6985
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op while saving (showing 5 of 6). These functions will not be directly callable after loading.
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
586/586 [=====] - 167s 285ms/step - loss: 0.9961 - accuracy: 0.6985 - val_loss: 0.9460 - val_accuracy: 0.7255 - lr: 0.0010
Epoch 24/70
586/586 [=====] - ETA: 0s - loss: 0.9955 - accuracy: 0.7022
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op while saving (showing 5 of 6). These functions will not be directly callable after loading.
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
586/586 [=====] - 171s 292ms/step - loss: 0.9955 - accuracy: 0.7022 - val_loss: 0.9618 - val_accuracy: 0.7316 - lr: 0.0010
Epoch 25/70
586/586 [=====] - 161s 275ms/step - loss: 0.9783 - accuracy: 0.7065 - val_loss: 0.9977 - val_accuracy: 0.7179 - lr: 0.0010
Epoch 26/70
586/586 [=====] - 165s 282ms/step - loss: 0.9768 - accuracy: 0.7076 - val_loss: 1.0056 - val_accuracy: 0.7144 - lr: 0.0010
Epoch 27/70
586/586 [=====] - ETA: 0s - loss: 0.9771 - accuracy: 0.7069
```

```
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op while saving (showing 5 of 6). These functions will not be directly callable after loading.  
INFO:tensorflow:Assets written to: output\cp.ckpt\assets  
INFO:tensorflow:Assets written to: output\cp.ckpt\assets  
586/586 [=====] - 167s 284ms/step - loss: 0.9771 - accuracy: 0.7069 - val_loss: 0.9131 - val_accuracy: 0.7432 - lr: 0.0010  
Epoch 28/70  
586/586 [=====] - 163s 278ms/step - loss: 0.9725 - accuracy: 0.7099 - val_loss: 0.9570 - val_accuracy: 0.7261 - lr: 0.0010  
Epoch 29/70  
586/586 [=====] - 167s 285ms/step - loss: 0.9718 - accuracy: 0.7109 - val_loss: 0.9883 - val_accuracy: 0.7163 - lr: 0.0010  
Epoch 30/70  
586/586 [=====] - ETA: 0s - loss: 0.9598 - accuracy: 0.7136  
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op while saving (showing 5 of 6). These functions will not be directly callable after loading.  
INFO:tensorflow:Assets written to: output\cp.ckpt\assets  
INFO:tensorflow:Assets written to: output\cp.ckpt\assets  
586/586 [=====] - 186s 317ms/step - loss: 0.9598 - accuracy: 0.7136 - val_loss: 0.9008 - val_accuracy: 0.7473 - lr: 0.0010  
Epoch 31/70  
586/586 [=====] - ETA: 0s - loss: 0.9582 - accuracy: 0.7166  
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op while saving (showing 5 of 6). These functions will not be directly callable after loading.  
INFO:tensorflow:Assets written to: output\cp.ckpt\assets  
INFO:tensorflow:Assets written to: output\cp.ckpt\assets  
586/586 [=====] - 193s 330ms/step - loss: 0.9582 - accuracy: 0.7166 - val_loss: 0.8629 - val_accuracy: 0.7560 - lr: 0.0010  
Epoch 32/70  
586/586 [=====] - ETA: 0s - loss: 0.9043 - accuracy: 0.7347  
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op while saving (showing 5 of 6). These functions will not be directly callable after loading.  
INFO:tensorflow:Assets written to: output\cp.ckpt\assets  
INFO:tensorflow:Assets written to: output\cp.ckpt\assets  
586/586 [=====] - 233s 397ms/step - loss: 0.9043 - accuracy: 0.7347 - val_loss: 0.8531 - val_accuracy: 0.7573 - lr: 5.0000e-04  
Epoch 33/70  
586/586 [=====] - ETA: 0s - loss: 0.8917 - accuracy: 0.786  
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op while saving (showing 5 of 6). These functions will not be directly callable after loading.  
INFO:tensorflow:Assets written to: output\cp.ckpt\assets  
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
```

```
586/586 [=====] - 257s 438ms/step - loss: 0.8917 - accuracy: 0.7386 - val_loss: 0.8483 - val_accuracy: 0.7593 - lr: 5.0000e-04
Epoch 34/70
586/586 [=====] - 250s 426ms/step - loss: 0.8811 - accuracy: 0.7422 - val_loss: 0.8879 - val_accuracy: 0.7465 - lr: 5.0000e-04
Epoch 35/70
586/586 [=====] - 294s 502ms/step - loss: 0.8732 - accuracy: 0.7443 - val_loss: 0.9261 - val_accuracy: 0.7373 - lr: 5.0000e-04
Epoch 36/70
586/586 [=====] - 162s 276ms/step - loss: 0.8658 - accuracy: 0.7424 - val_loss: 0.8739 - val_accuracy: 0.7568 - lr: 5.0000e-04
Epoch 37/70
586/586 [=====] - ETA: 0s - loss: 0.8600 - accuracy: 0.7465
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op while saving (showing 5 of 6). These functions will not be directly callable after loading.
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
586/586 [=====] - 171s 292ms/step - loss: 0.8600 - accuracy: 0.7465 - val_loss: 0.8299 - val_accuracy: 0.7690 - lr: 5.0000e-04
Epoch 38/70
586/586 [=====] - 174s 296ms/step - loss: 0.8577 - accuracy: 0.7445 - val_loss: 0.9066 - val_accuracy: 0.7446 - lr: 5.0000e-04
Epoch 39/70
586/586 [=====] - 179s 305ms/step - loss: 0.8523 - accuracy: 0.7486 - val_loss: 0.8197 - val_accuracy: 0.7686 - lr: 5.0000e-04
Epoch 40/70
586/586 [=====] - 174s 298ms/step - loss: 0.8438 - accuracy: 0.7524 - val_loss: 0.8855 - val_accuracy: 0.7462 - lr: 5.0000e-04
Epoch 41/70
586/586 [=====] - 173s 295ms/step - loss: 0.8426 - accuracy: 0.7516 - val_loss: 0.8220 - val_accuracy: 0.7675 - lr: 5.0000e-04
Epoch 42/70
586/586 [=====] - ETA: 0s - loss: 0.8430 - accuracy: 0.7497
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op while saving (showing 5 of 6). These functions will not be directly callable after loading.
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
586/586 [=====] - 177s 302ms/step - loss: 0.8430 - accuracy: 0.7497 - val_loss: 0.8138 - val_accuracy: 0.7731 - lr: 5.0000e-04
Epoch 43/70
586/586 [=====] - ETA: 0s - loss: 0.8358 - accuracy: 0.7523
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op while saving (showing 5 of 6). These functions will not be directly callable after loading.
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
```

```
586/586 [=====] - 171s 293ms/step - loss: 0.8358 - accuracy: 0.7523 - val_loss: 0.7513 - val_accuracy: 0.7900 - lr: 5.0000e-04
Epoch 44/70
586/586 [=====] - 175s 299ms/step - loss: 0.8298 - accuracy: 0.7540 - val_loss: 0.8204 - val_accuracy: 0.7672 - lr: 5.0000e-04
Epoch 45/70
586/586 [=====] - 180s 306ms/step - loss: 0.8302 - accuracy: 0.7517 - val_loss: 0.7936 - val_accuracy: 0.7754 - lr: 5.0000e-04
Epoch 46/70
586/586 [=====] - 172s 294ms/step - loss: 0.8227 - accuracy: 0.7567 - val_loss: 0.8651 - val_accuracy: 0.7542 - lr: 5.0000e-04
Epoch 47/70
586/586 [=====] - ETA: 0s - loss: 0.8239 - accuracy: 0.7571
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op while saving (showing 5 of 6). These functions will not be directly callable after loading.
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
586/586 [=====] - 172s 294ms/step - loss: 0.8239 - accuracy: 0.7571 - val_loss: 0.7414 - val_accuracy: 0.7913 - lr: 5.0000e-04
Epoch 48/70
586/586 [=====] - 184s 314ms/step - loss: 0.8254 - accuracy: 0.7571 - val_loss: 0.7677 - val_accuracy: 0.7837 - lr: 5.0000e-04
Epoch 49/70
586/586 [=====] - 185s 316ms/step - loss: 0.8184 - accuracy: 0.7559 - val_loss: 0.8371 - val_accuracy: 0.7622 - lr: 5.0000e-04
Epoch 50/70
586/586 [=====] - 179s 305ms/step - loss: 0.8176 - accuracy: 0.7567 - val_loss: 0.8496 - val_accuracy: 0.7582 - lr: 5.0000e-04
Epoch 51/70
586/586 [=====] - 182s 311ms/step - loss: 0.8149 - accuracy: 0.7573 - val_loss: 0.7883 - val_accuracy: 0.7802 - lr: 5.0000e-04
Epoch 52/70
586/586 [=====] - 169s 288ms/step - loss: 0.7961 - accuracy: 0.7644 - val_loss: 0.7423 - val_accuracy: 0.7904 - lr: 3.0000e-04
Epoch 53/70
586/586 [=====] - 171s 291ms/step - loss: 0.7866 - accuracy: 0.7685 - val_loss: 0.8700 - val_accuracy: 0.7574 - lr: 3.0000e-04
Epoch 54/70
586/586 [=====] - 167s 286ms/step - loss: 0.7816 - accuracy: 0.7696 - val_loss: 0.7921 - val_accuracy: 0.7781 - lr: 3.0000e-04
Epoch 55/70
586/586 [=====] - 176s 301ms/step - loss: 0.7748 - accuracy: 0.7690 - val_loss: 0.7591 - val_accuracy: 0.7850 - lr: 3.0000e-04
Epoch 56/70
586/586 [=====] - 179s 305ms/step - loss: 0.7716 - accuracy: 0.7712 - val_loss: 0.7837 - val_accuracy: 0.7790 - lr: 3.0000e-04
Epoch 57/70
586/586 [=====] - 177s 302ms/step - loss: 0.7719 - accuracy: 0.7703 - val_loss: 0.7577 - val_accuracy: 0.7828 - lr: 3.0000e-04
Epoch 57: early stopping
```

```
In [ ]: keras.models.load_model('output\\cp.ckpt')
```

```
Out[ ]: <keras.engine.sequential.Sequential at 0x2085d2fa320>
```

## Training history visualization

```
In [ ]: def plot_history(history):
    # Plot training & validation accuracy values
```

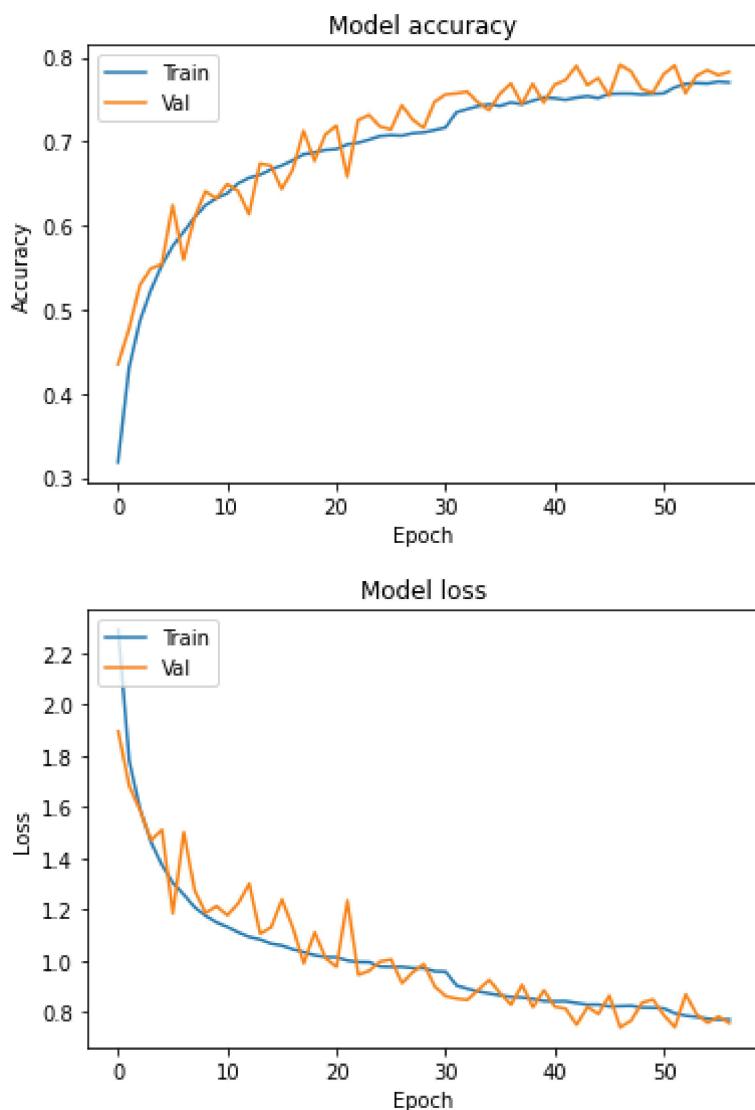
```

plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Model accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train', 'Val'], loc='upper left')
plt.show()

# Plot training & validation Loss values
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Val'], loc='upper left')
plt.show()

```

In [ ]: `plot_history(history)`



## Evaluate the model

In [ ]:

```

_, train_acc = model.evaluate(x_train, y_train, verbose=1)
_, val_acc = model.evaluate(x_val, y_val, verbose=1)

_, test_acc = model.evaluate(x_test, y_test, verbose=1)
print('Train: %.3f, Val: %.3f, Test: %.3f' % (train_acc, val_acc, test_acc))

```

1172/1172 [=====] - 46s 40ms/step - loss: 0.6684 - accuracy: 0.8117  
391/391 [=====] - 16s 41ms/step - loss: 0.7577 - accuracy: 0.7828  
313/313 [=====] - 13s 40ms/step - loss: 0.7716 - accuracy: 0.7810  
Train: 0.812, val: 0.783, Test: 0.781