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, ra
```

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

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)
```

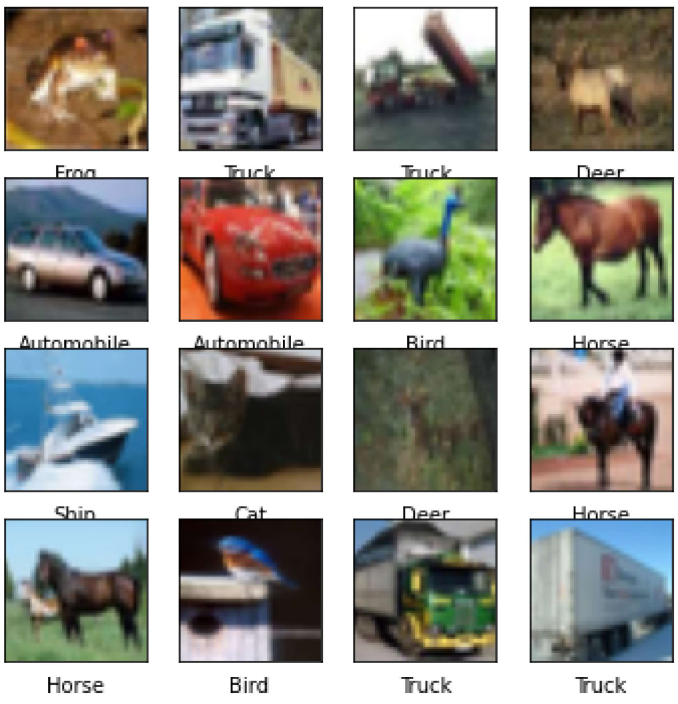
```
print("Size of the test set")
print("x_test", x_test.shape)
print("y_test", y_test.shape)
```

```
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 #
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

Layer (type)	Output Shape	Param #
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 (MaxPooling  (None, 4, 4, 128)        0
2D)

dropout_2 (Dropout)          (None, 4, 4, 128)        0

flatten (Flatten)            (None, 2048)             0

dense (Dense)                (None, 10)               20490

=================================================================
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 > 75:
        lrate = 0.0005
    if epoch > 100:
        lrate = 0.0003
    return lrate

model.compile(optimizer=rms,
              loss=loss[0],
              metrics=[metrics[0]],
              )
```

In [ ]:
```
batch_size = 64
epochs = 50
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=Tr
```

```
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,cp_
```

```
Epoch 1/50
586/586 [==============================] - ETA: 0s - loss: 2.3242 - accuracy: 0.3070
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op, _jit_comp
iled_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 d
irectly callable after loading.
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
586/586 [==============================] - 129s 218ms/step - loss: 2.3242 - accuracy:
0.3070 - val_loss: 1.9493 - val_accuracy: 0.4092 - lr: 0.0010
Epoch 2/50
586/586 [==============================] - ETA: 0s - loss: 1.8087 - accuracy: 0.4149
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op, _jit_comp
iled_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 d
irectly callable after loading.
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
586/586 [==============================] - 158s 269ms/step - loss: 1.8087 - accuracy:
0.4149 - val_loss: 1.9111 - val_accuracy: 0.4114 - lr: 0.0010
Epoch 3/50
586/586 [==============================] - ETA: 0s - loss: 1.5818 - accuracy: 0.4805
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op, _jit_comp
iled_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 d
irectly callable after loading.
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
586/586 [==============================] - 157s 268ms/step - loss: 1.5818 - accuracy:
0.4805 - val_loss: 1.4974 - val_accuracy: 0.5237 - lr: 0.0010
Epoch 4/50
586/586 [==============================] - ETA: 0s - loss: 1.4645 - accuracy: 0.5170
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op, _jit_comp
iled_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 d
irectly callable after loading.
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
586/586 [==============================] - 161s 276ms/step - loss: 1.4645 - accuracy:
0.5170 - val_loss: 1.3880 - val_accuracy: 0.5632 - lr: 0.0010
Epoch 5/50
586/586 [==============================] - ETA: 0s - loss: 1.3652 - accuracy: 0.5447
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op, _jit_comp
iled_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 d
irectly callable after loading.
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
```

```
586/586 [==============================] - 160s 273ms/step - loss: 1.3652 - accuracy:
0.5447 - val_loss: 1.3271 - val_accuracy: 0.5739 - lr: 0.0010
Epoch 6/50
586/586 [==============================] - ETA: 0s - loss: 1.3001 - accuracy: 0.5710
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op, _jit_comp
iled_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 d
irectly callable after loading.
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
586/586 [==============================] - 163s 278ms/step - loss: 1.3001 - accuracy:
0.5710 - val_loss: 1.3256 - val_accuracy: 0.5870 - lr: 0.0010
Epoch 7/50
586/586 [==============================] - ETA: 0s - loss: 1.2492 - accuracy: 0.5894
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op, _jit_comp
iled_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 d
irectly callable after loading.
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
586/586 [==============================] - 160s 273ms/step - loss: 1.2492 - accuracy:
0.5894 - val_loss: 1.2002 - val_accuracy: 0.6286 - lr: 0.0010
Epoch 8/50
586/586 [==============================] - ETA: 0s - loss: 1.2060 - accuracy: 0.6093
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op, _jit_comp
iled_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 d
irectly callable after loading.
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
586/586 [==============================] - 163s 278ms/step - loss: 1.2060 - accuracy:
0.6093 - val_loss: 1.1642 - val_accuracy: 0.6438 - lr: 0.0010
Epoch 9/50
586/586 [==============================] - 159s 271ms/step - loss: 1.1763 - accuracy:
0.6237 - val_loss: 1.2640 - val_accuracy: 0.6058 - lr: 0.0010
Epoch 10/50
586/586 [==============================] - ETA: 0s - loss: 1.1465 - accuracy: 0.6312
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op, _jit_comp
iled_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 d
irectly callable after loading.
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
586/586 [==============================] - 161s 275ms/step - loss: 1.1465 - accuracy:
0.6312 - val_loss: 1.0966 - val_accuracy: 0.6635 - lr: 0.0010
Epoch 11/50
586/586 [==============================] - ETA: 0s - loss: 1.1284 - accuracy: 0.6403
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op, _jit_comp
iled_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 d
irectly callable after loading.
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
```

```
586/586 [==============================] - 165s 282ms/step - loss: 1.1284 - accuracy:
0.6403 - val_loss: 1.0844 - val_accuracy: 0.6723 - lr: 0.0010
Epoch 12/50
586/586 [==============================] - 159s 271ms/step - loss: 1.1015 - accuracy:
0.6488 - val_loss: 1.1750 - val_accuracy: 0.6532 - lr: 0.0010
Epoch 13/50
586/586 [==============================] - 159s 272ms/step - loss: 1.0863 - accuracy:
0.6585 - val_loss: 1.1223 - val_accuracy: 0.6611 - lr: 0.0010
Epoch 14/50
586/586 [==============================] - ETA: 0s - loss: 1.0674 - accuracy: 0.6653
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op, _jit_comp
iled_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 d
irectly callable after loading.
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
586/586 [==============================] - 163s 277ms/step - loss: 1.0674 - accuracy:
0.6653 - val_loss: 1.0033 - val_accuracy: 0.7006 - lr: 0.0010
Epoch 15/50
586/586 [==============================] - 150s 256ms/step - loss: 1.0535 - accuracy:
0.6714 - val_loss: 1.0621 - val_accuracy: 0.6839 - lr: 0.0010
Epoch 16/50
586/586 [==============================] - 149s 254ms/step - loss: 1.0440 - accuracy:
0.6767 - val_loss: 1.1641 - val_accuracy: 0.6570 - lr: 0.0010
Epoch 17/50
586/586 [==============================] - 150s 256ms/step - loss: 1.0332 - accuracy:
0.6809 - val_loss: 1.1352 - val_accuracy: 0.6666 - lr: 0.0010
Epoch 18/50
586/586 [==============================] - 150s 256ms/step - loss: 1.0216 - accuracy:
0.6848 - val_loss: 1.1449 - val_accuracy: 0.6621 - lr: 0.0010
Epoch 19/50
586/586 [==============================] - ETA: 0s - loss: 1.0171 - accuracy: 0.6905
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op, _jit_comp
iled_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 d
irectly callable after loading.
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
586/586 [==============================] - 152s 260ms/step - loss: 1.0171 - accuracy:
0.6905 - val_loss: 1.0192 - val_accuracy: 0.7081 - lr: 0.0010
Epoch 20/50
586/586 [==============================] - 147s 251ms/step - loss: 1.0089 - accuracy:
0.6932 - val_loss: 1.0587 - val_accuracy: 0.6930 - lr: 0.0010
Epoch 21/50
586/586 [==============================] - 147s 251ms/step - loss: 1.0006 - accuracy:
0.6946 - val_loss: 1.1206 - val_accuracy: 0.6800 - lr: 0.0010
Epoch 22/50
586/586 [==============================] - 165s 281ms/step - loss: 0.9973 - accuracy:
0.6982 - val_loss: 1.1797 - val_accuracy: 0.6639 - lr: 0.0010
Epoch 23/50
586/586 [==============================] - ETA: 0s - loss: 0.9817 - accuracy: 0.7020
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op, _jit_comp
iled_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 d
irectly callable after loading.
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
```

```
586/586 [==============================] - 163s 278ms/step - loss: 0.9817 - accuracy:
0.7020 - val_loss: 0.9160 - val_accuracy: 0.7387 - lr: 0.0010
Epoch 24/50
586/586 [==============================] - ETA: 0s - loss: 0.9824 - accuracy: 0.7046
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op, _jit_comp
iled_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 d
irectly 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: 0.9824 - accuracy:
0.7046 - val_loss: 0.8817 - val_accuracy: 0.7493 - lr: 0.0010
Epoch 25/50
586/586 [==============================] - 263s 448ms/step - loss: 0.9790 - accuracy:
0.7063 - val_loss: 0.9680 - val_accuracy: 0.7185 - lr: 0.0010
Epoch 26/50
586/586 [==============================] - 210s 358ms/step - loss: 0.9701 - accuracy:
0.7110 - val_loss: 0.9322 - val_accuracy: 0.7382 - lr: 0.0010
Epoch 27/50
586/586 [==============================] - 163s 279ms/step - loss: 0.9660 - accuracy:
0.7109 - val_loss: 0.9172 - val_accuracy: 0.7404 - lr: 0.0010
Epoch 28/50
586/586 [==============================] - 250s 427ms/step - loss: 0.9608 - accuracy:
0.7152 - val_loss: 1.1947 - val_accuracy: 0.6725 - lr: 0.0010
Epoch 29/50
586/586 [==============================] - 232s 395ms/step - loss: 0.9560 - accuracy:
0.7167 - val_loss: 0.8962 - val_accuracy: 0.7482 - lr: 0.0010
Epoch 30/50
586/586 [==============================] - 189s 322ms/step - loss: 0.9539 - accuracy:
0.7156 - val_loss: 0.9997 - val_accuracy: 0.7143 - lr: 0.0010
Epoch 31/50
586/586 [==============================] - ETA: 0s - loss: 0.9450 - accuracy: 0.7215
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op, _jit_comp
iled_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 d
irectly callable after loading.
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
586/586 [==============================] - 198s 338ms/step - loss: 0.9450 - accuracy:
0.7215 - val_loss: 0.8846 - val_accuracy: 0.7573 - lr: 0.0010
Epoch 32/50
586/586 [==============================] - 186s 317ms/step - loss: 0.9421 - accuracy:
0.7223 - val_loss: 0.9226 - val_accuracy: 0.7398 - lr: 0.0010
Epoch 33/50
586/586 [==============================] - ETA: 0s - loss: 0.9486 - accuracy: 0.7203
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op, _jit_comp
iled_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 d
irectly callable after loading.
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
INFO:tensorflow:Assets written to: output\cp.ckpt\assets
```

```
586/586 [==============================] - 194s 331ms/step - loss: 0.9486 - accuracy:
0.7203 - val_loss: 0.8478 - val_accuracy: 0.7599 - lr: 0.0010
Epoch 34/50
586/586 [==============================] - 555s 949ms/step - loss: 0.9436 - accuracy:
0.7218 - val_loss: 0.9180 - val_accuracy: 0.7440 - lr: 0.0010
Epoch 35/50
586/586 [==============================] - 191s 326ms/step - loss: 0.9409 - accuracy:
0.7223 - val_loss: 0.9255 - val_accuracy: 0.7398 - lr: 0.0010
Epoch 36/50
586/586 [==============================] - 8017s 14s/step - loss: 0.9400 - accuracy:
0.7260 - val_loss: 0.9381 - val_accuracy: 0.7407 - lr: 0.0010
Epoch 37/50
586/586 [==============================] - 194s 331ms/step - loss: 0.9354 - accuracy:
0.7261 - val_loss: 0.9375 - val_accuracy: 0.7376 - lr: 0.0010
Epoch 38/50
586/586 [==============================] - 170s 289ms/step - loss: 0.9370 - accuracy:
0.7273 - val_loss: 0.9444 - val_accuracy: 0.7402 - lr: 0.0010
Epoch 39/50
586/586 [==============================] - 138s 235ms/step - loss: 0.9309 - accuracy:
0.7289 - val_loss: 0.9442 - val_accuracy: 0.7394 - lr: 0.0010
Epoch 40/50
586/586 [==============================] - 234s 399ms/step - loss: 0.9286 - accuracy:
0.7283 - val_loss: 0.9604 - val_accuracy: 0.7314 - lr: 0.0010
Epoch 41/50
586/586 [==============================] - 280s 477ms/step - loss: 0.9254 - accuracy:
0.7311 - val_loss: 0.9528 - val_accuracy: 0.7370 - lr: 0.0010
Epoch 42/50
586/586 [==============================] - 276s 471ms/step - loss: 0.9223 - accuracy:
0.7310 - val_loss: 0.8763 - val_accuracy: 0.7566 - lr: 0.0010
Epoch 43/50
586/586 [==============================] - 278s 474ms/step - loss: 0.9212 - accuracy:
0.7310 - val_loss: 0.8862 - val_accuracy: 0.7567 - lr: 0.0010
Epoch 43: early stopping
```

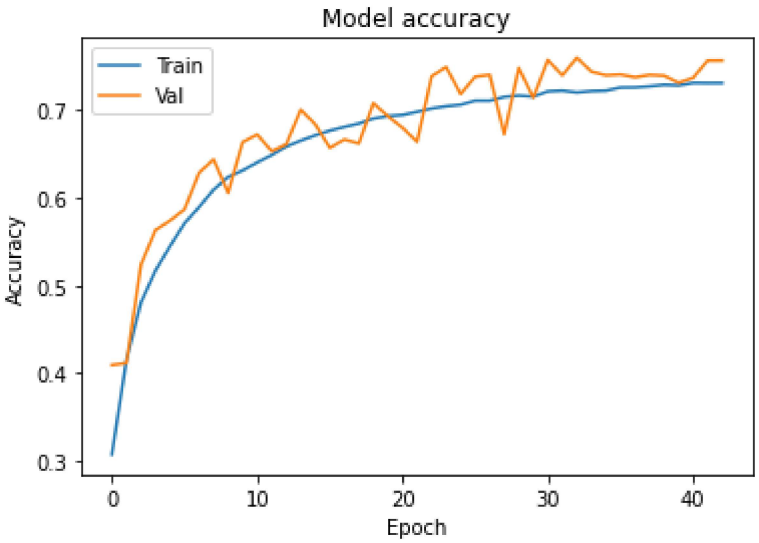
In [ ]:
```
keras.models.load_model('output\\cp.ckpt')
```

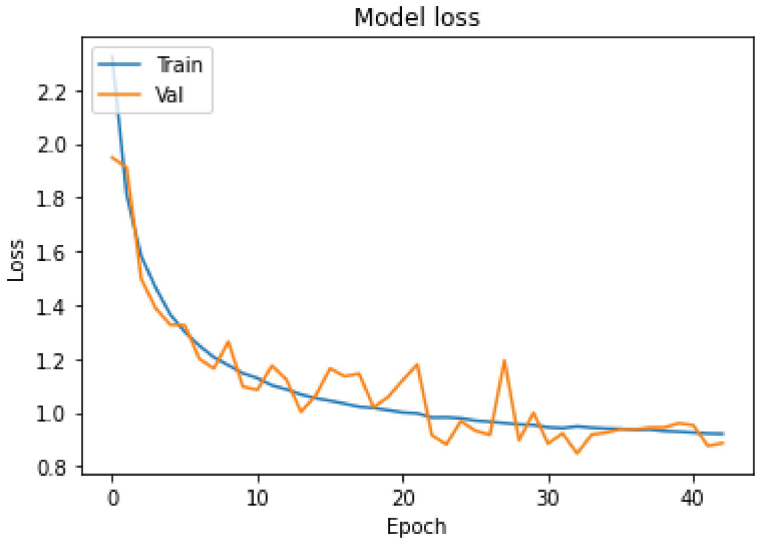
Training history visualization

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 [==============================] - 72s 62ms/step - loss: 0.8200 - accuracy:
0.7703
391/391 [==============================] - 19s 49ms/step - loss: 0.8862 - accuracy:
0.7567
313/313 [==============================] - 16s 51ms/step - loss: 0.9013 - accuracy:
0.7488
Train: 0.770, val: 0.757,Test: 0.749
```