

Image Classification using CNN on CIFAR-10 Dataset

```
In [1]: import keras
        from keras.datasets import cifar10
        from keras.preprocessing.image import ImageDataGenerator
        from keras.models import Sequential
        from keras.layers import Dense, Dropout, Activation, Flatten
        from keras.layers import Conv2D, MaxPooling2D
        import matplotlib.pyplot as plt
        import os
```

Using TensorFlow backend.

```
In [2]: batch_size = 32
        num_classes = 10
        epochs = 10
        num_predictions = 20
        save_dir = os.path.join(os.getcwd(), 'saved_models')
        model_name = 'keras_cifar10_trained_model.h5'
```

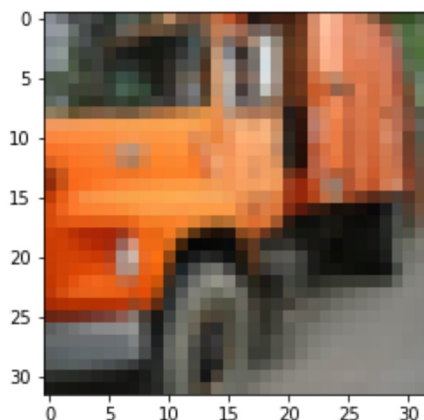
```
In [3]: (x_train, y_train), (x_test, y_test) = cifar10.load_data()
```

```
In [4]: print('x_train shape:', x_train.shape)
        print(x_train.shape[0], 'train samples')
        print(x_test.shape[0], 'test samples')
```

x_train shape: (50000, 32, 32, 3)
50000 train samples
10000 test samples

```
In [5]: plt.imshow(x_train[6666])
```

Out[5]: <matplotlib.image.AxesImage at 0x17ac22ff780>



```
In [6]: y_train = keras.utils.to_categorical(y_train, num_classes)
        y_test = keras.utils.to_categorical(y_test, num_classes)
```

```
In [7]: model = Sequential()
model.add(Conv2D(32, (3, 3), padding='same',
               input_shape=x_train.shape[1:]))
model.add(Activation('relu'))
model.add(Conv2D(32, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
```

WARNING:tensorflow:From d:\ProgramData\Anaconda3\lib\site-packages\tensorflow\python\framework\op_def_library.py:263: colocate_with (from tensorflow.python.framework.ops) is deprecated and will be removed in a future version.

Instructions for updating:

Colocations handled automatically by placer.

WARNING:tensorflow:From d:\ProgramData\Anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:3445: calling dropout (from tensorflow.python.ops.nn_ops) with keep_prob is deprecated and will be removed in a future version.

Instructions for updating:

Please use `rate` instead of `keep_prob`. Rate should be set to `rate = 1 - keep_prob`.

```
In [8]: model.add(Conv2D(64, (3, 3), padding='same'))
model.add(Activation('relu'))
model.add(Conv2D(64, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
```

```
In [9]: model.add(Flatten())
model.add(Dense(512))
model.add(Activation('relu'))
model.add(Dropout(0.5))
model.add(Dense(num_classes))
model.add(Activation('softmax'))
```

```
In [10]: opt = keras.optimizers.rmsprop(lr=0.0001, decay=1e-6)
```

```
In [11]: model.compile(loss='categorical_crossentropy',
                      optimizer=opt,
                      metrics=['accuracy'])

x_train = x_train.astype('float32')
x_test = x_test.astype('float32')
x_train /= 255
x_test /= 255
```

```
In [40]: print('Not using data augmentation.')
model.fit(x_train, y_train,
          batch_size=batch_size,
          epochs=epochs,
          validation_data=(x_test, y_test),
          shuffle=True)
```

Not using data augmentation.

Train on 50000 samples, validate on 10000 samples

Epoch 1/10

50000/50000 [=====] - 112s 2ms/step - loss: 1.8486 - acc: 0.3246 - val_loss: 1.5515 - val_acc: 0.4357

Epoch 2/10

50000/50000 [=====] - 112s 2ms/step - loss: 1.5273 - acc: 0.4441 - val_loss: 1.3738 - val_acc: 0.5077

Epoch 3/10

50000/50000 [=====] - 113s 2ms/step - loss: 1.3915 - acc: 0.4997 - val_loss: 1.2625 - val_acc: 0.5479

Epoch 4/10

50000/50000 [=====] - 113s 2ms/step - loss: 1.2941 - acc: 0.5388 - val_loss: 1.2480 - val_acc: 0.5532

Epoch 5/10

50000/50000 [=====] - 113s 2ms/step - loss: 1.2210 - acc: 0.5664 - val_loss: 1.1591 - val_acc: 0.5863

Epoch 6/10

50000/50000 [=====] - 114s 2ms/step - loss: 1.1530 - acc: 0.5930 - val_loss: 1.1260 - val_acc: 0.5992

Epoch 7/10

50000/50000 [=====] - 110s 2ms/step - loss: 1.1007 - acc: 0.6122 - val_loss: 1.0084 - val_acc: 0.6435

Epoch 8/10

50000/50000 [=====] - 113s 2ms/step - loss: 1.0465 - acc: 0.6306 - val_loss: 1.0335 - val_acc: 0.6333

Epoch 9/10

50000/50000 [=====] - 113s 2ms/step - loss: 0.9999 - acc: 0.6512 - val_loss: 0.9826 - val_acc: 0.6591

Epoch 10/10

50000/50000 [=====] - 113s 2ms/step - loss: 0.9553 - acc: 0.6660 - val_loss: 0.8840 - val_acc: 0.6954

Out[40]: <keras.callbacks.History at 0x1e4a0afdd68>

```
In [12]: if not os.path.isdir(save_dir):
          os.makedirs(save_dir)
          model_path = os.path.join(save_dir, model_name)
          model.save(model_path)
          print('Saved trained model at %s ' % model_path)
```

Saved trained model at C:\Users\Project\Desktop\Workshop\saved_models\keras_cifar10_trained_model.h5

```
In [13]: scores = model.evaluate(x_test, y_test)
          print('Test loss:', scores[0])
          print('Test accuracy:', scores[1])
```

10000/10000 [=====] - 8s 820us/step

Test loss: 2.3041815254211424

Test accuracy: 0.1029