

# Convolutional Neural Networks for Image Classification

## Abstract

Recognizing photos is one of the most common problems, and it is also important in image search engines such as Google or AliExpress where the user uses image contents to query for similar things.

The goal of this project is to build a deep learning model that can recognize objects in images and predict the class of the image.

## Design

The project is based on CNN model, I will build a convolution neural network in Keras on a CIFAR-10 dataset, which consists of 60,000 with the size of 32 by 32 pixels color images in 10 classes.

## Data

CIFAR-10 dataset consists of 60,000 32x32 pixels color images in 10 classes. The 10 different classes represent airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks, in which each of those classes consists of 6000 images.

CIFAR-10 dataset is already available in the datasets module of Keras you can directly import it from `keras.datasets` or download it from [here](#).

## Algorithms

The Training dataset consists of 50,000 32X32 color images of 10 different objects.

- Convert the pixel values of the dataset to float type.
- Create the sequential model and add the layers (Convolution layer, batchNormalization layer, pooling layer, dropout layer, flattening layer and Dense Layers)
- Fit and evaluate the model

## Tools

- NumPy for data manipulation.
- TensorFlow, Keras, Scikit-learn for modeling.
- Matplotlib for visualizations.

## Communication

<matplotlib.legend.Legend at 0x232ae2d0ec8>

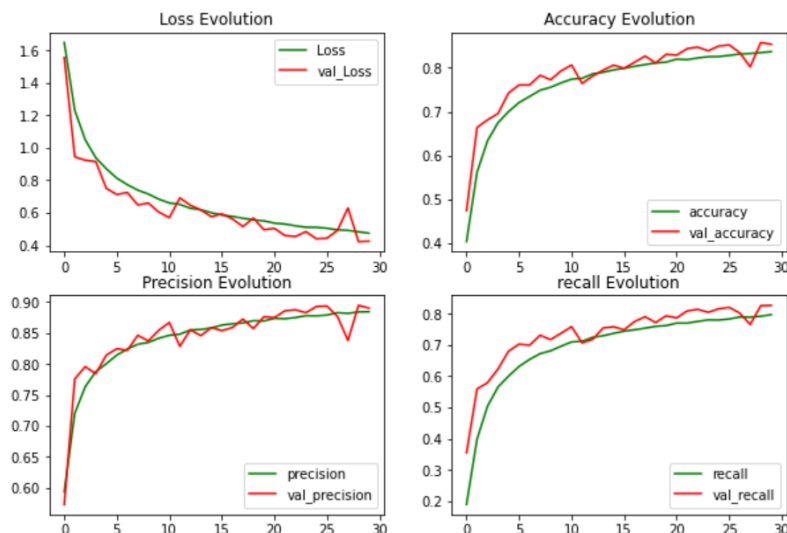


Figure 1: Evolution model

```
evaluation = model.evaluate(x_test, y_test, verbose=0)
print(f'Test Accuracy : {evaluation[1] * 100:.2f}%')
```

Test Accuracy : 85.39%

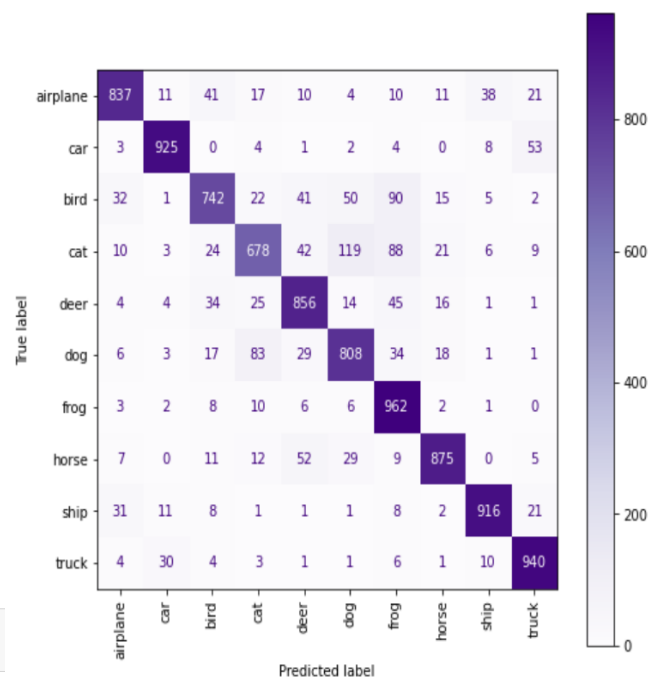


Figure 2: Heatmap

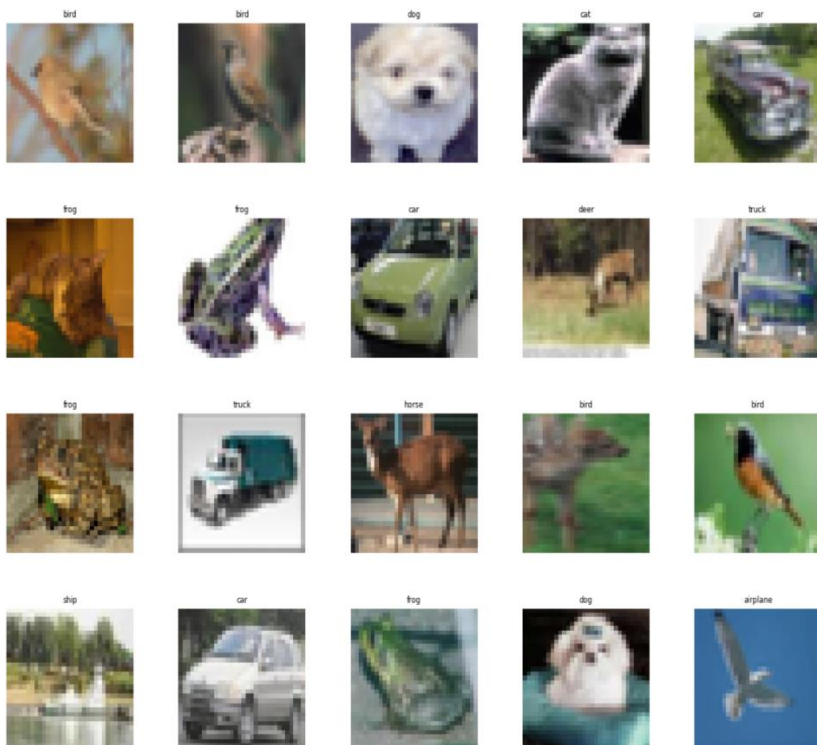


Figure 3: Testing

```
img2 = image.load_img("car.jpg", target_size=(32,32))
image_to_test = image.img_to_array(img2) / 255
list_of_images = np.expand_dims(image_to_test, axis=0)

result = model.predict(list_of_images)

predict = result[0]

labels_class = int(np.argmax(predict))

plt.imshow(img2)

<matplotlib.image.AxesImage at 0x232a6a82a88>

0
5
10
15
20
25
30

class_likelihood = predict[labels_class]
class_label = labels[labels_class]
print("This is image is a {} - likelihood: {:.2f}".format(class_label, class_likelihood))

This is image is a car - likelihood: 0.613107
```