## СОЗДАНИЕ НЕЙРОННОЙ СЕТИ

Андреев Вячеслав, г. Петрозаводск Детский технопарк Кванториум «Сампо», ІТквантум Наставник: Иван Ригоев

## Моя нейронная сеть будет отличать Стрекоз от Бабочек Её нужно обучить по фотографиям. У меня их получилось 1408

```
# Каталог с данными для обучения
train dir = 'train'
# Каталог с данными для проверки
val dir = 'val'
# Каталог с данными для тестирования
test dir = 'test'
# Размеры изображения
img width, img height = 150, 150
# Размерность тензора на основе изображения для входных данных в нейронную сеть
# backend Tensorflow, channels last
input shape = (img width, img height, 3)
# Размер мини-выборки
batch size = 64
# Количество изображений для обучения
nb train samples = 590
# Количество изображений для проверки
nb_validation_samples = 398
# Количество изображений для тестирования
nb test samples = 420
```

Эта часть кода, отвечает за изображения для обучения, проверки и тестирования

Layer (type)	Output Shape	Param #
input_2 (InputLayer)	(None, 150, 150, 3)	0
block1_conv1 (Conv2D)	(None, 150, 150, 64)	1792
block1_conv2 (Conv2D)	(None, 150, 150, 64)	36928
block1_pool (MaxPooling2D)	(None, 75, 75, 64)	0
block2_conv1 (Conv2D)	(None, 75, 75, 128)	73856
block2_conv2 (Conv2D)	(None, 75, 75, 128)	147584
block2_pool (MaxPooling2D)	(None, 37, 37, 128)	0
block3_conv1 (Conv2D)	(None, 37, 37, 256)	295168
block3_conv2 (Conv2D)	(None, 37, 37, 256)	590080
block3_conv3 (Conv2D)	(None, 37, 37, 256)	590080
block3_pool (MaxPooling2D)	(None, 18, 18, 256)	0
block4_conv1 (Conv2D)	(None, 18, 18, 512)	1180160
block4_conv2 (Conv2D)	(None, 18, 18, 512)	2359808
block4_conv3 (Conv2D)	(None, 18, 18, 512)	2359808
block4_pool (MaxPooling2D)	(None, 9, 9, 512)	0
block5_conv1 (Conv2D)	(None, 9, 9, 512)	2359808
block5_conv2 (Conv2D)	(None, 9, 9, 512)	2359808
block5_conv3 (Conv2D)	(None, 9, 9, 512)	2359808
block5_pool (MaxPooling2D)	(None, 4, 4, 512)	0

Total params: 14,714,688 Trainable params: 0

Non-trainable params: 14,714,688

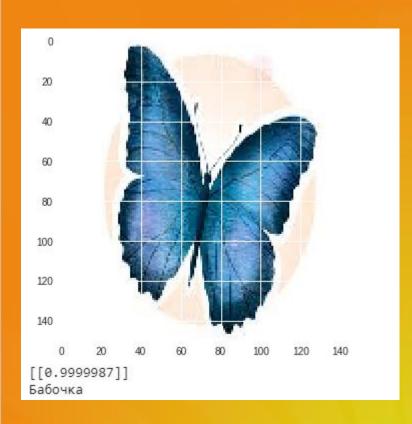
## А,эта часть кода, отвечает за вид сети и количество слоев в ней

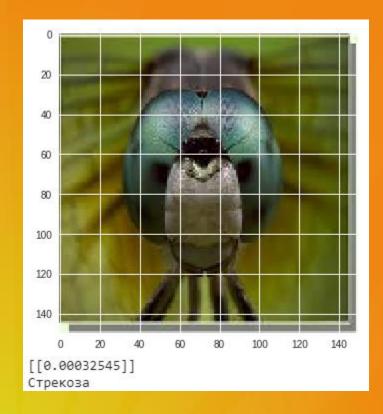
```
Epoch 1/10
Epoch 2/10
Epoch 3/10
Epoch 4/10
Epoch 5/10
Epoch 6/10
Epoch 7/10
Epoch 8/10
Epoch 9/10
Epoch 10/10
<tensorflow.python.keras.callbacks.History at 0x7fe054816f60>
scores = model.evaluate generator(test generator, 50)
print("Аккуратность на тестовых данных: %.2f%%" % (scores[1]*100))
Аккуратность на тестовых данных: 98.26%
                 Обучение нашей сети
```

Точность тестовых данных

```
import numpy as np
import smtplib
from keras.applications.vgg16 import preprocess input
import matplotlib.pyplot as plt
from tensorflow.python.keras.preprocessing import image
img = image.load_img('ButterFly.jpg', target_size=(150, 150))
plt.imshow(img)
plt.show()
x = image.img to array(img)
x = np.expand_dims(x, axis=0)
x = preprocess input(x)
prediction = model.predict(x)
print(prediction)
if prediction > 0.5:
  print("Бабочка")
  print("Стрекоза")
```

Эта часть различает бабочек и стрекоз с помощью обученной нейронной сети





После загрузки картинки мы видим результат.