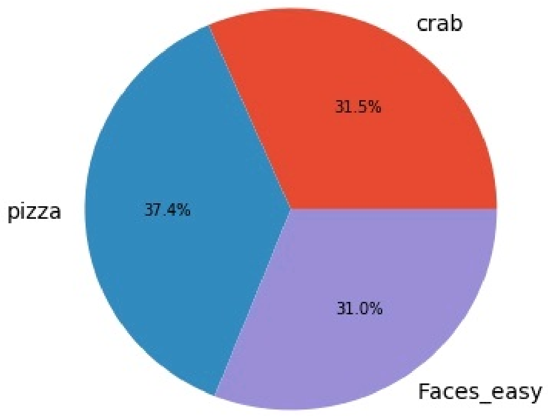
Image Classifier Multi-Label

By: Mácio Matheus Santos de Arruda

Multilabel classification using convolutional networks. This project uses the dataset Caltech 101, a famous database for computer vision applications.

The dataset

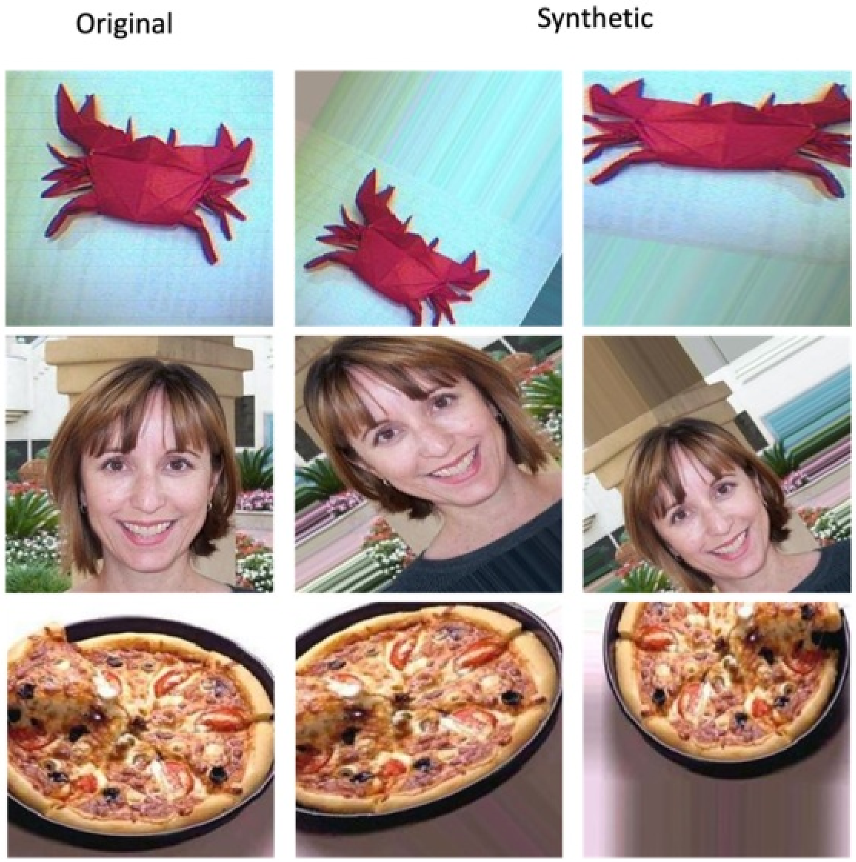
We selected 3 different classes of the dataset caltech101: Crab; Faces_Easy and Pizza. Below is a graph indicating the percentage of each:



The data augmentation

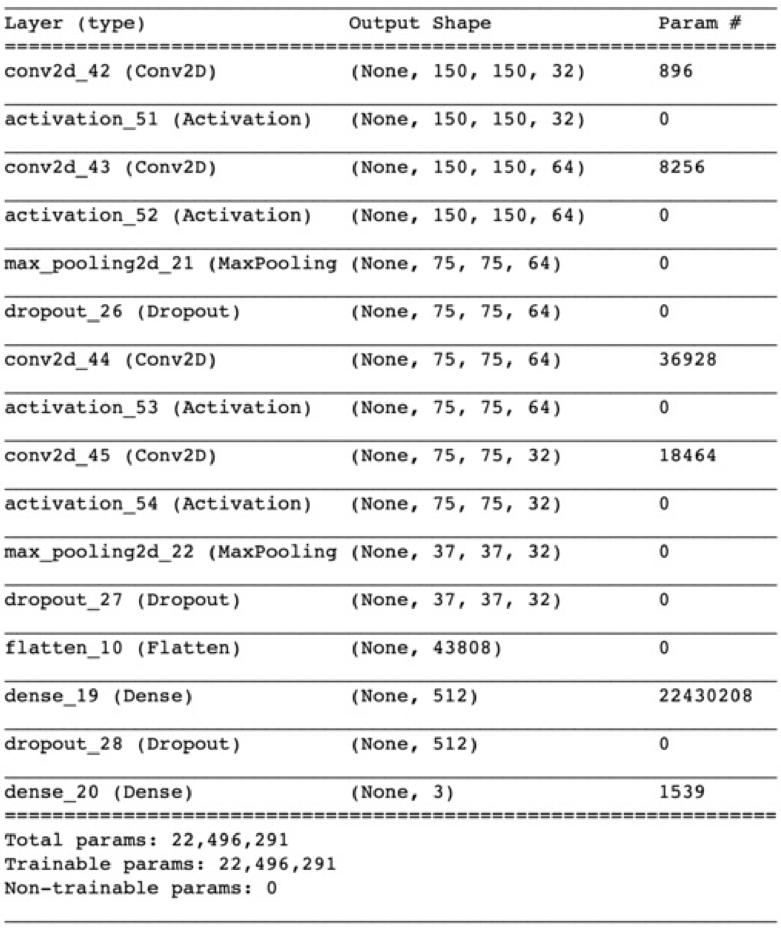
An increase of database was done, to work better with convolutional network.

Five images were generated for each image in the dataset. Below is an example of an increase made for an instance of each class.



The CNN Summary

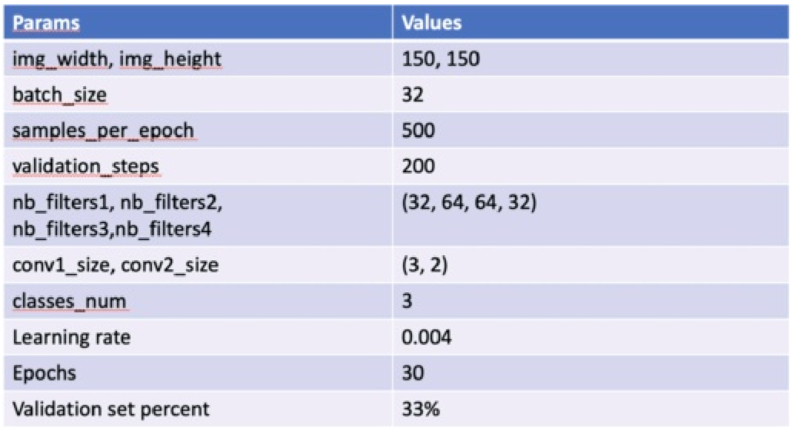
Using Keras, the neural network below was assembled to solve the presented classification problem:



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_42 (Conv2D)           (None, 150, 150, 32)      896
_________________________________________________________________
activation_51 (Activation)   (None, 150, 150, 32)      0
_________________________________________________________________
conv2d_43 (Conv2D)           (None, 150, 150, 64)      8256
_________________________________________________________________
activation_52 (Activation)   (None, 150, 150, 64)      0
_________________________________________________________________
max_pooling2d_21 (MaxPooling (None, 75, 75, 64)        0
_________________________________________________________________
dropout_26 (Dropout)         (None, 75, 75, 64)        0
_________________________________________________________________
conv2d_44 (Conv2D)           (None, 75, 75, 64)        36928
_________________________________________________________________
activation_53 (Activation)   (None, 75, 75, 64)        0
_________________________________________________________________
conv2d_45 (Conv2D)           (None, 75, 75, 32)        18464
_________________________________________________________________
activation_54 (Activation)   (None, 75, 75, 32)        0
_________________________________________________________________
max_pooling2d_22 (MaxPooling (None, 37, 37, 32)        0
_________________________________________________________________
dropout_27 (Dropout)         (None, 37, 37, 32)        0
_________________________________________________________________
flatten_10 (Flatten)         (None, 43808)             0
_________________________________________________________________
dense_19 (Dense)             (None, 512)               22430208
_________________________________________________________________
dropout_28 (Dropout)         (None, 512)               0
_________________________________________________________________
dense_20 (Dense)             (None, 3)                 1539
=================================================================
Total params: 22,496,291
Trainable params: 22,496,291
Non-trainable params: 0
_________________________________________________________________
```

Parameterization

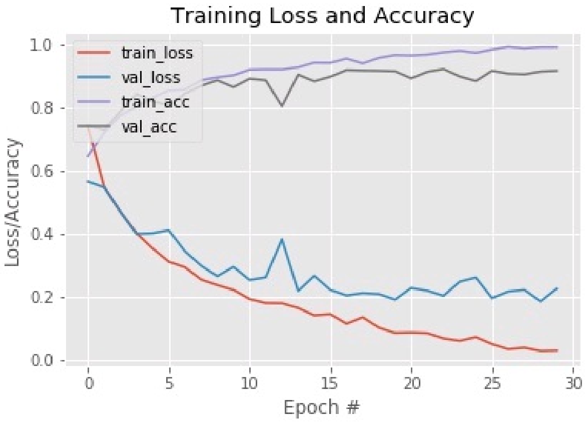
Parameterization used in the model



Params	Values
img_width, img_height	150, 150
batch_size	32
samples_per_epoch	500
validation_steps	200
nb_filters1, nb_filters2, nb_filters3,nb_filters4	(32, 64, 64, 32)
conv1_size, conv2_size	(3, 2)
classes_num	3
Learning rate	0.004
Epochs	30
Validation set percent	33%

Results Loss and Accuracy

Below is the graph of loss and accuracy in training and validation sets during training



Test live

Prediction performed on images that were outside the training set and validation. The crabs were intentionally pointed in circle format to cause error in the model:

Usage

First of all, build the container using docker-compose and then you can access the Jupyter that is ready to be used.

Run with docker compose

```
cd computer-vision
docker-compose up -d
```

Accessing Jupyter

```
http://<your-ip>:8111/tree
```

Ports

```
    - 8888 => Jupyter
    - 6011 => Tensorboard
    - 5011 => App
```

DockerHub

```
https://hub.docker.com/r/maciomatheus/jupyter_notebook_data_science/
```