



## Project 07 – Fried Eggs

### Fried Eggs Quality Assessment

*Summary:* This project is an introduction to Deep Learning and computer vision  
(CNN, instance segmentation, object detection, data labeling)

*Version:* 1.00

# Contents

I	Preamble	2
II	Introduction	3
III	Goals	4
IV	General instructions	5
V	Mandatory part	6
V.1	Task . . . . .	6
V.2	Dataset . . . . .	6
V.3	Implementation . . . . .	7
V.3.1	Data Labeling . . . . .	7
V.3.2	Data Preprocessing . . . . .	8
V.3.3	Classification . . . . .	8
V.4	Submission . . . . .	9
VI	Bonus part	10
VII	Submission and peer-evaluation	11

# Chapter I

## Preamble

Teaching machines to see – what can be more incredible? But before teaching somebody to do something we need to understand it at first. No understanding – no teaching. It is not an extraordinary fact that the first breakthrough in computer vision happened after researchers had found out in 1959 how animals (to be specific, cats) perceive visual information.

During an experiment one of the neurons fired when the researchers were slipping a new slide into the projector. The neuron was activated by “the movement of the line created by the shadow of the sharp edge of the glass slide”. Later on they realized that there are simple and complex neurons and the processing of visual information starts from simple edges and shapes.

In 1982 another researcher found out that we at first see simple edges, then we see simple shapes, then we see how these shapes are organized in more complex shapes and structures. It means that processing of visual information is hierarchical. It became the core concept of deep learning.

A little later convolutional layers started being used in computer vision. The main idea of them is that they are basically rectangular receptive fields that have inside a weight vector (filter). They slide on a picture and create a new representation of it through this convolutional operation.

In 1989 Yann LeCun applied a backprop style learning algorithm that you are already familiar with to convolutional architecture of neural nets. Since that time it has become a part of the machine learning field. Before that, researchers tried to create those filters by themselves. But still in terms of quality CNNs (convolutional neural nets) could not achieve something better than the other approaches and humans themselves.

Only in 2012 a new architecture (AlexNet) was applied to the famous ImageNet competition and showed the best result. Since that time CNNs have become the mainstream in computer vision and deep learning. Approximately during that time a committee of CNNs showed a better-than-human accuracy in traffic signs recognition.

Nowadays, computer vision is actively used in healthcare, agriculture, entertainment and automotive fields.

# Chapter II

## Introduction

There are several popular tasks in computer vision: image classification, detection and semantic segmentation. What is the difference?

In image classification tasks we try to get an answer on the question “Does this object exist on the image?”. We do not care where exactly. In detection tasks we care about it and the answer that we would like to get is a bounding box around that object. In semantic segmentation tasks we still care about the position and we want to get information about each pixel of the image – which object does it belong to? Classification is a simpler task and segmentation is harder. But one way or another in all of them CNNs are widely used.

The central element of CNNs is the kernel.

It “looks” at the image having its own point of view (weights inside) by sliding over the image and making simple calculations: element-wise product of matrices and sum of the results. You can see the example on the picture above. By doing this it creates a so-called feature map. That can go to the next convolutional layer or through other processing – for example, pooling. The role of pooling is to decrease the size of the image. It helps the net to work with more high-level features and also increases the speed of training. There are different kinds of pooling. For example, max pooling just takes the maximum element in the frame of the receptive field. Average pooling takes the average of those elements, etc.

The last layers of a CNN are fully-connected layers from a classical neural net architecture that you are already familiar with. The output from the last convolutional layer is being flattened and sent to a sequence of FCN layers the output of which is a layer with the number of neurons equal to the number of classes of your classification task.

That is how CNNs work in simple words. Of course, there are more sophisticated architecture and techniques, but you will get to know them in your own journey of solving the project.

# **Chapter III**

## **Goals**

The goal of this project is to give you a first approach to Deep Learning and Computer Vision. You will try to help a restaurant to maintain good quality of their fried eggs. In order to do this you will also have to work with data labeling tools and approaches.

# Chapter IV

## General instructions

- This project will only be evaluated by humans. You are free to organize and name your files as you desire.
- Here and further we use Python 3 as the only correct version of Python.
- You can use any framework (TensorFlow, PyTorch, etc.) that you may find useful.
- The norm is not applied to this project. Nevertheless, you are asked to be clear and structured in the conception of your source code.
- Store the datasets in the subfolder **data**.

# Chapter V

## Mandatory part

### V.1 Task

In this project, you will try to analyze photos of fried eggs made in a restaurant to make a judgement whether these fried eggs are good enough to be served or not. There can be different problems in the cooking process: too few yolks, too many of them, some of them might be broken, it can be overcooked.

Here are the tasks that you need to do:

- 1. data labeling with instance segmentation,
- 2. data preprocessing,
- 3. fried eggs classification.

### V.2 Dataset

You will work with the [dataset](#) of fried eggs. It contains 199 unlabeled images of fried eggs (train) and 65 images from the hidden test sample for which we know classes.



The classes are the following:

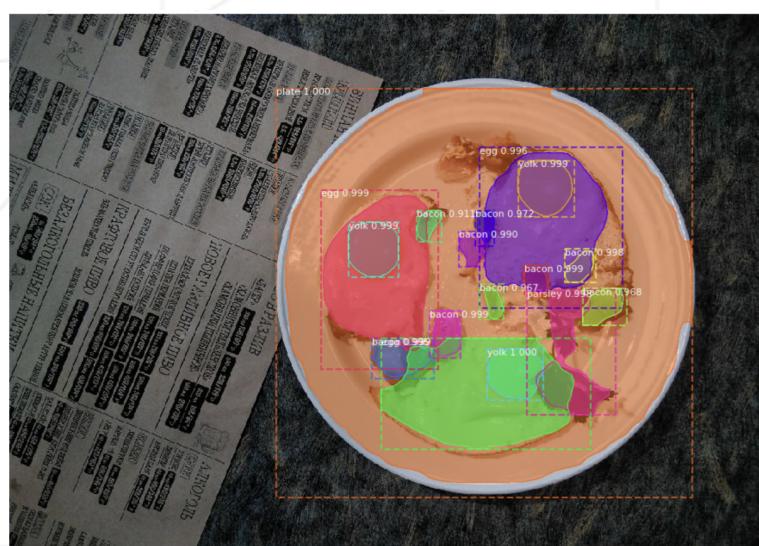
- 0 – good fried eggs (three yolks, parsley in the center, well-cooked, looks like a whole and not several separate pieces),
- 1 – overcooked or overturned fried eggs,
- 2 – fried eggs with two yolks,
- 3 – fried eggs with a broken yolk,
- 4 – fried eggs with four yolks,
- 5 – fried eggs with a wrong position of ingredients or they are missing, wrong placement or composition, undercooked.

## V.3 Implementation

You can work in [Google Colab](#) or Jupyter Notebooks on your computer.

### V.3.1 Data Labeling

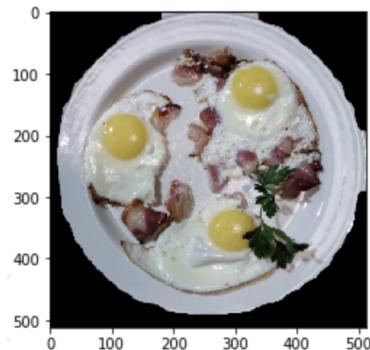
- 1. You need to label the images from the train sample using 6 classes described above.
- 2. In order to solve the project with a good quality you will have to solve the instance segmentation task. To make it happen you need to annotate your images with instances of different objects: to show where on the image you have a piece of bacon, yolk or a piece of parsley. Inference annotations as masks to polygons (XMLs)



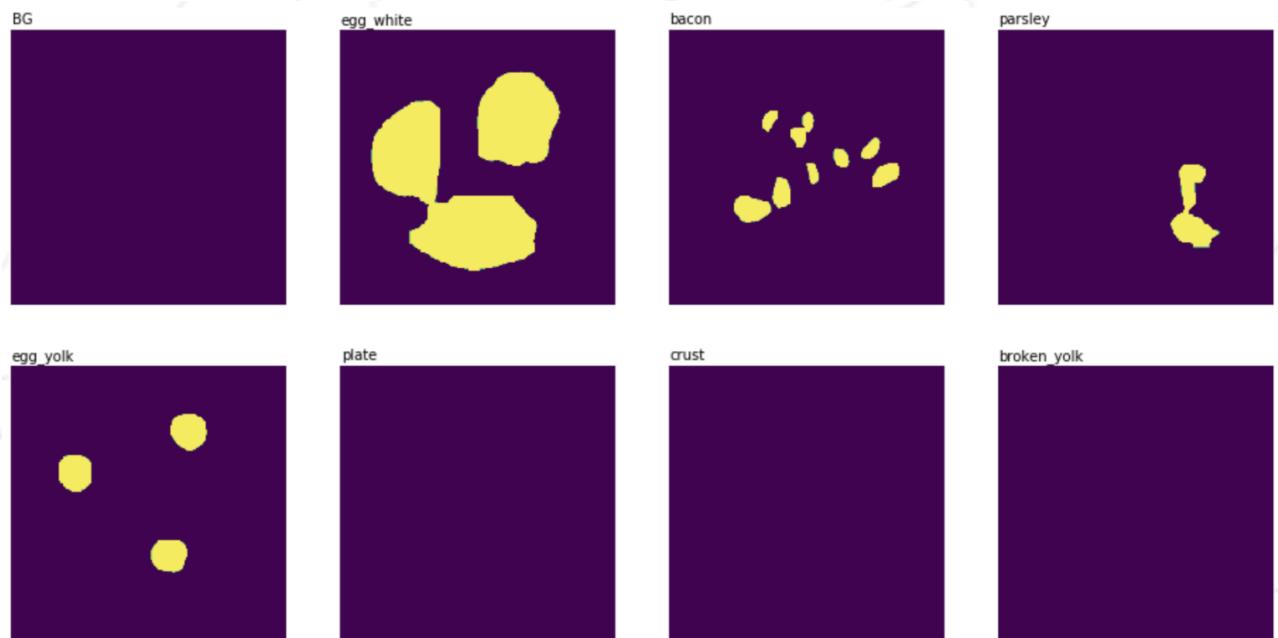
- 3. To automate this process use the active learning approach by training a neural net that will learn from you which objects are on the image.

### V.3.2 Data Preprocessing

- 1. Resize images to 1024x768.
- 2. You need to learn to crop images: only a plate with fried eggs should go to a CNN.



- 3. Extract masks of the ingredients. Use them as an input to CNN.



### V.3.3 Classification

- 1. Use raw images, masks, the number of different ingredients as input for a CNN.
- 2. Try different approaches and algorithms. You will need to achieve logloss on the test dataset at most – 9.56.

## V.4 Submission

Your repository should contain one or several notebooks with your solutions and visualizations. It should contain a CSV file with the list of filenames and corresponding classes. You can download the template from [here](#).

# **Chapter VI**

## **Bonus part**

- 1. Work with data augmentation.
- 2. Achieve a better logloss – 2.66.

# **Chapter VII**

## **Submission and peer-evaluation**

Submit your work on your Git repository as usual. Only the work on your repository will be graded.

Here are the points that your peer-corrector will have to check:

- if images are labeled using classes and instance segmentation,
- if active learning approach was used to automate data labeling,
- if images were resized, cropped,
- if masks were extracted from segmentation,
- if the required logloss was achieved in the classification.