

Дипломный проект

Разработка модели распознавания качества продукта

Зеленин Дмитрий Викторович

Профессия Data Scientist группа DS-10

Машинное обучение: фундаментальные инструменты и практики AML-8

Москва 2021 г.

1. Постановка задачи

С усовершенствованием методов и алгоритмов машинного обучения искусственного интеллекта ряд коммерческих задач бизнеса переходит в цифровой мир аналитических вычислений и предсказаний. Главная цель бизнеса, - качество предоставляемого продукта, как фактор узнаваемости и популярности бренда. Как следствие рост прибыли, - одного из главных показателей эффективности компании.

В рамках данного проекта был предложен для анализа датасет с отчётами фотографий тайных покупателей пиццы известной российской сети ресторанов быстрого питания.

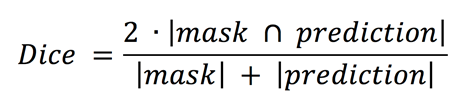
В качестве исходных данных был предоставлен комбинированный фотоотчёт в виде ссылочного файла на фотографии в интернете в формате \*.csv. Кроме служебной информации в файле были предоставлены фотографии пицц объёмом 300-500 kb в различных ракурсах и указанием на наличие обнаруженных дефектов перечнем более 40 позиций. Дополнительно было сказано, что некоторых фоток может и не быть.

Надо выбрать дефект и выявить алгоритм для определения наличия этого дефекта на фотографиях пиццы.

Следует сразу сказать, что среди перечня обнаруженных дефектов, с которыми можно будет ознакомиться в п.2 этого отчёта, был выбран дефект «горелый борт». В качестве инструмента по обнаружению «горелых бортов» на пицце была использована свёрточная нейронная сеть U-Net для семантической сегментации предоставленных фотографий и дальнейшего предсказания наличия или отсутствия горелых бортов.

Метрикой, которая служит для обучения сети, является коэффициент Дайса [1] (Dice Coefficient) (так же называется коэффициент Сёренсена – Sorensen-Dice coefficient) или Жаккара (Jaccard similarity coefficient), который показывает меру сходства – в данном случае, показывающий меру площади правильно отмеченных сегментов (отношение площади пересечения к площади объединения).

Языком формул коэффициент Дайса выглядит следующим образом:

,

где в качестве площадей служат меры области маски и области предсказанной маски.

1. Анализ

Со времени первого представления свёрточной сети U-Net [2] в рамках семантической сегментации медицинских снимков в 2015 году данная сеть стала очень популярной и широко применяется в датасетах с минимальным количеством тренировочных данных.

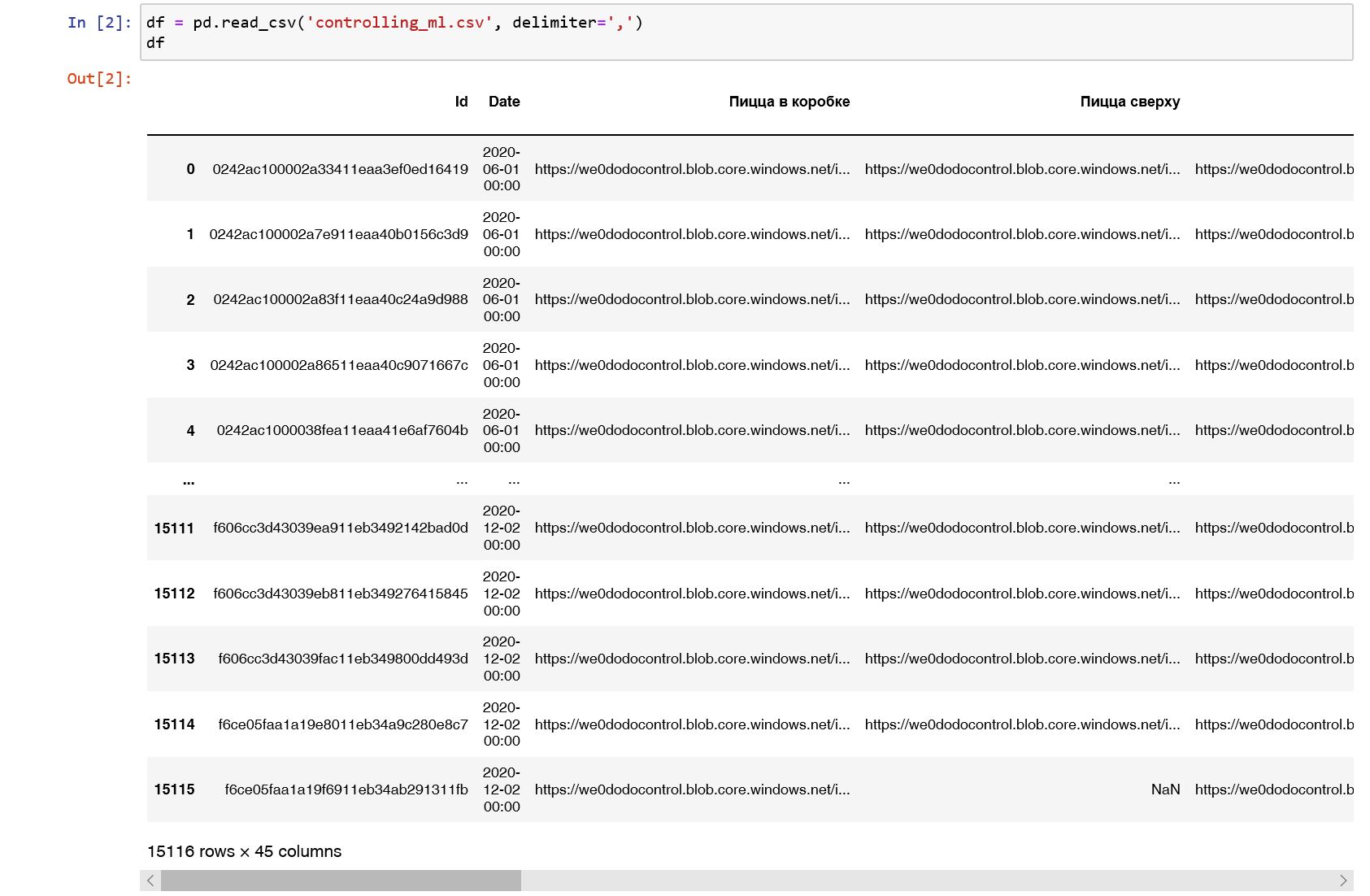
Например:

* классификация нервов с помощью ультразвука на [соревновании](https://www.kaggle.com/c/ultrasound-nerve-segmentation) Kaggle, несколько примеров ядер и образцов кода: [1](https://github.com/jocicmarko/ultrasound-nerve-segmentation) [2](https://www.kaggle.com/zfturbo/keras-is-there-any-nerve) [3](https://github.com/yihui-he/u-net) [4](https://github.com/petrosgk/Kaggle-Carvana-Image-Masking-Challenge);
* статья и код по [сегментации кровеносных сосудов сетчатки](https://github.com/orobix/retina-unet);
* еще одна [реализация](https://github.com/zizhaozhang/unet-tensorflow-keras) U-net с Keras;
* применение небольшой сети U-net для [распознавания автомобилей](https://chatbotslife.com/small-u-net-for-vehicle-detection-9eec216f9fd6).

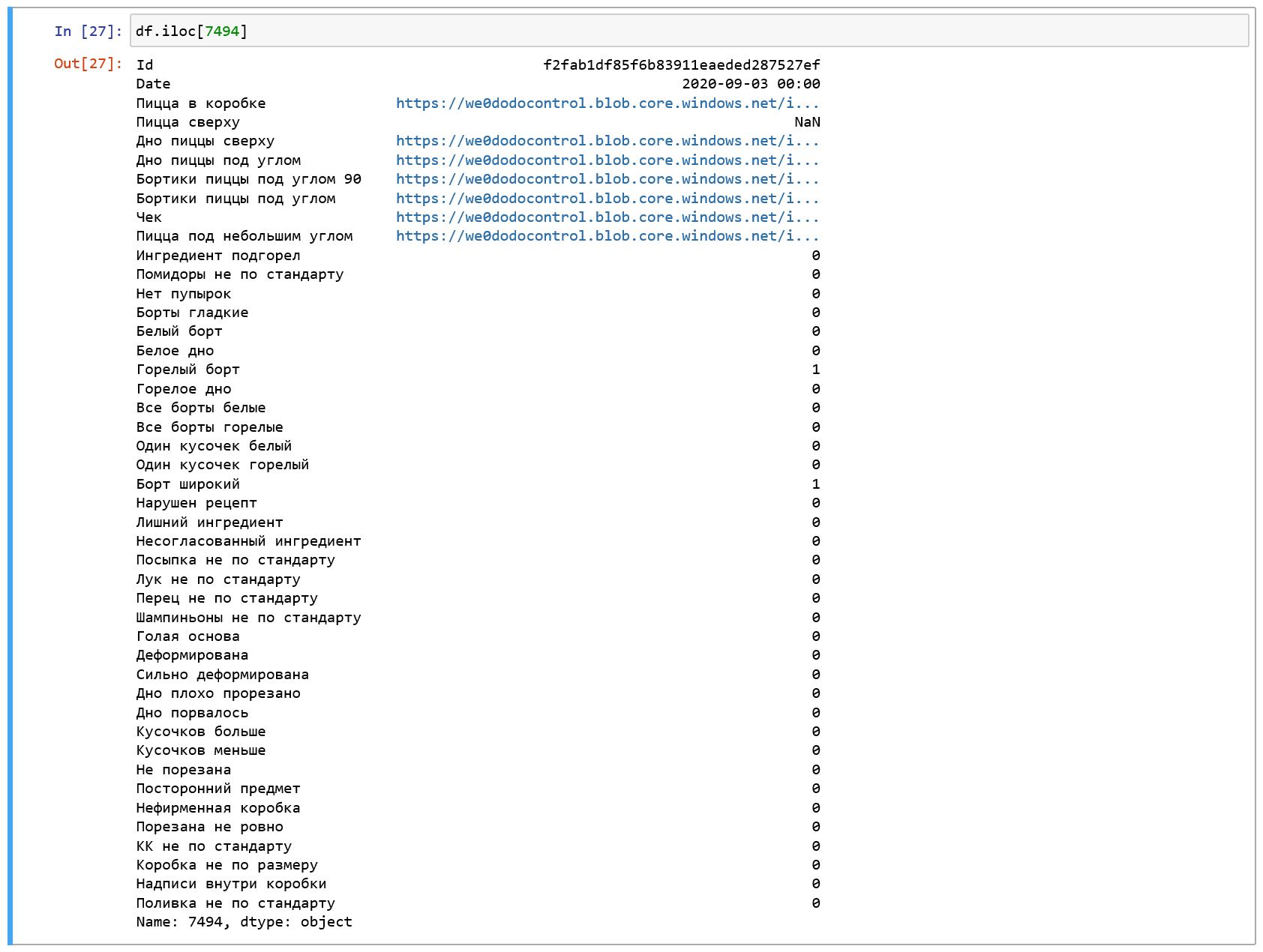
Безусловно, есть предобученные модели, с известными весовыми коэффициентами (обучение с учителем). В рамках данного дипломного проекта представилась задача по обучению свёрточной сети U-Net с ранее не размеченными данными.

Проведём предварительный анализ данных.

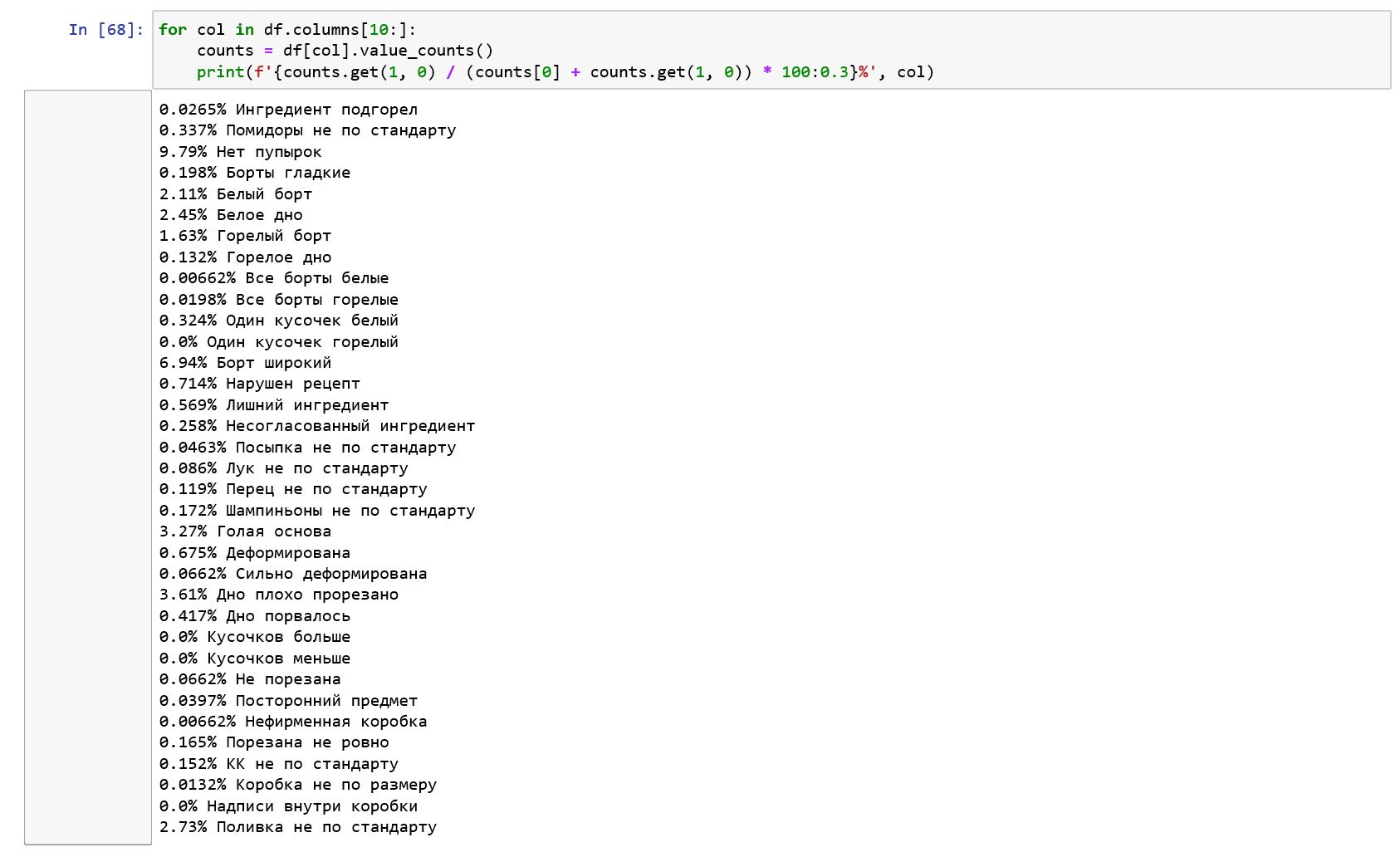
Выведем данные в виде таблицы с помощью программы Jupyter Notebook из предоставленного файла ’controlling\_ml.csv’.



Покажем, что из себя представляет данные в таблице на примере произвольных данных:



Определим соотношение количество дефектов к общему числу данных по категориям:

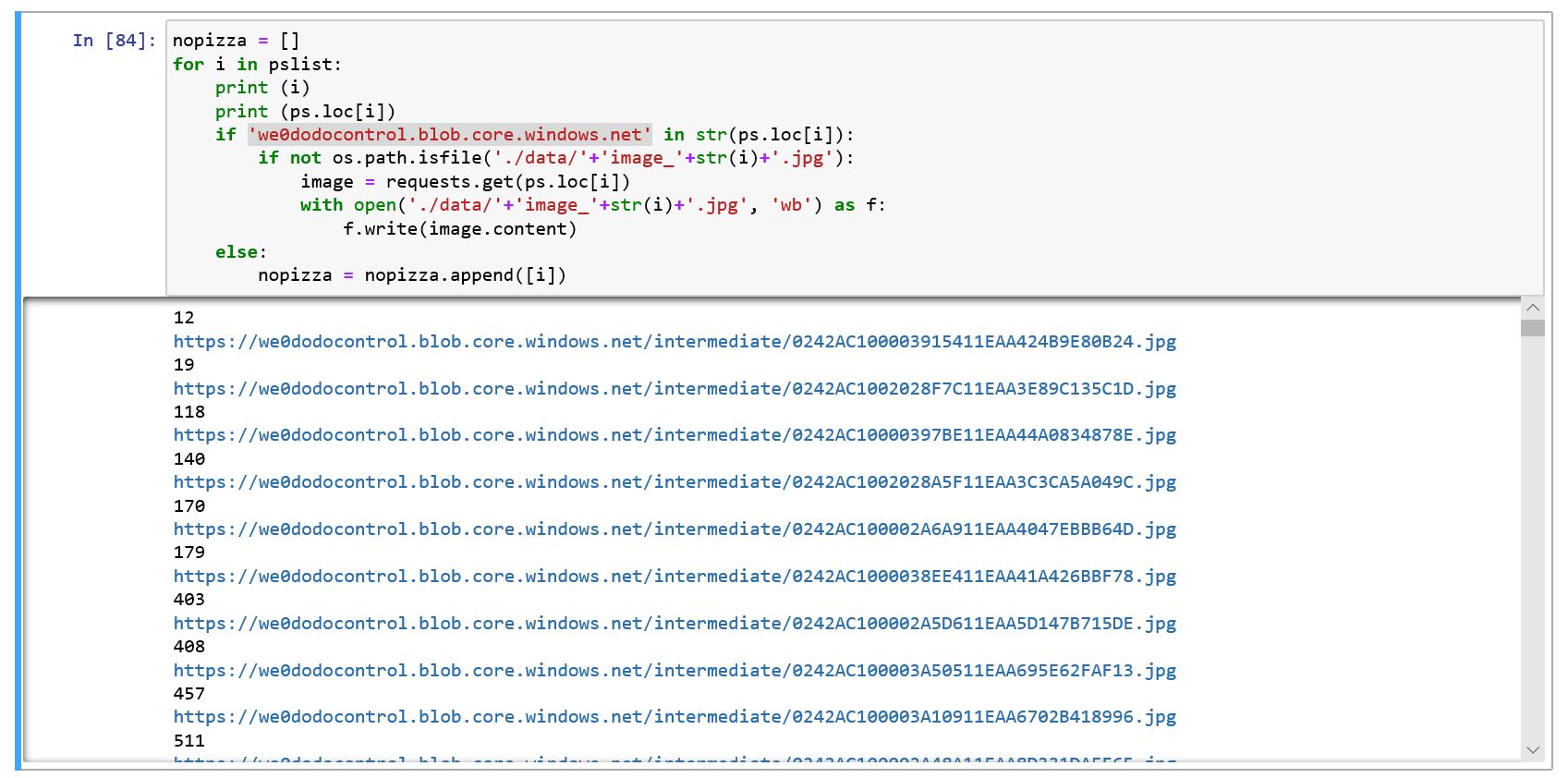


Создадим список с интернет-адресами на фотки с дефектом «горелый борт»:



Естественно, здесь могут быть пустые значения.

Проведём request запрос, предполагая, что электронный адрес содержит строку 'we0dodocontrol.blob.core.windows.net', тогда точно уйдём от проблем пустых записей.

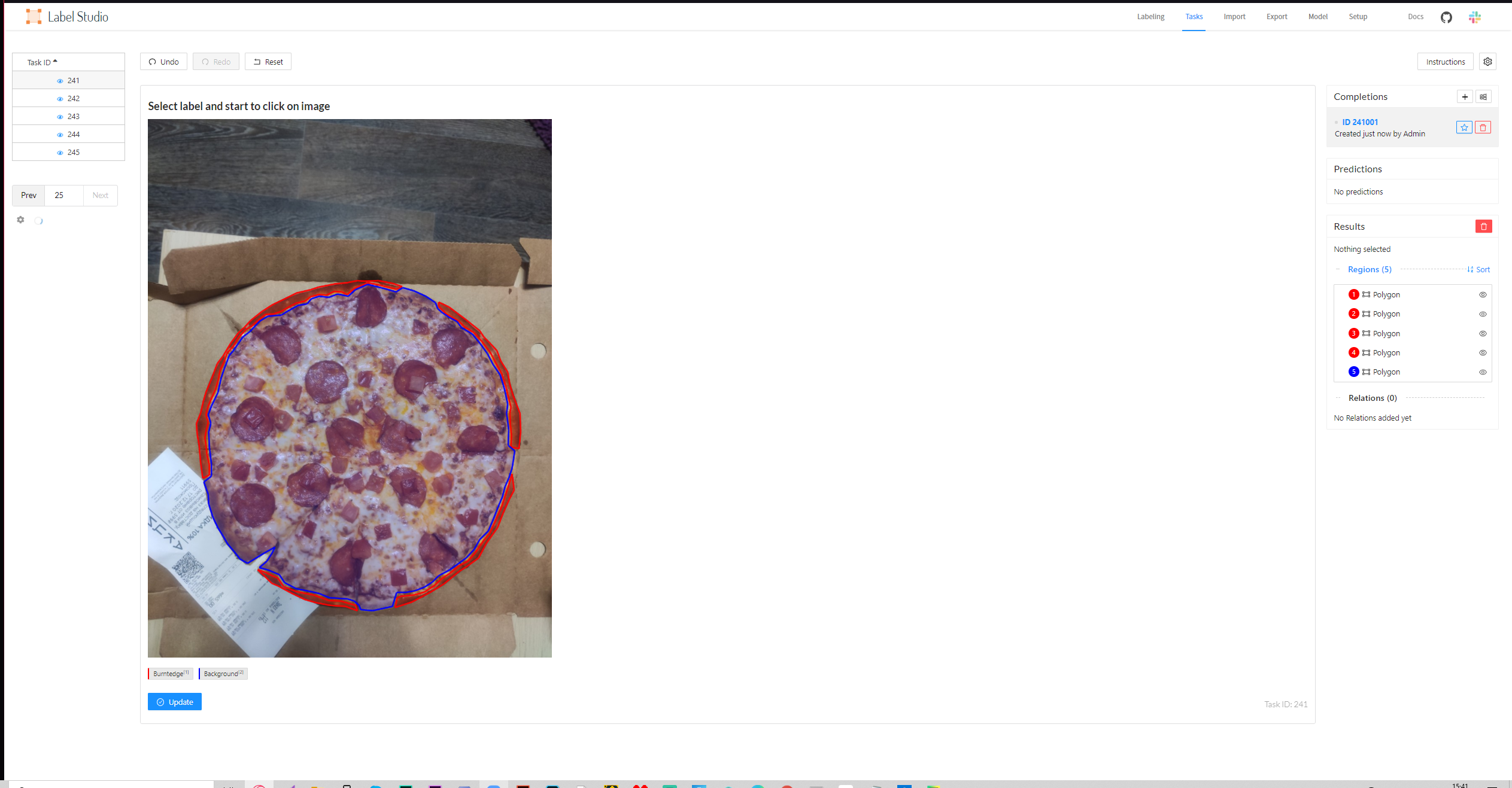


Таким образом, мы загрузили первичный датасет фотографий, которые и будем скармливать нейронной сети U-Net, чтобы научиться определять горелые края на пицце.

1. Методика решения

3.1 Разметка изображений в программе LabelStudio

Проведена разметка изображений вручную в программе LabelStudio [3]. Всего было размечено 245 полученных на этапе предварительного анализа данных изображений.



* 1. Приведение изображения и маски к одному размеру

Для работы свёрточной сети U-Net на вход требуется совпадения размеров маски и изображения, без учёта их совместного изменения размера на вход сети. Поскольку программа LabelStudio выдаёт размер маски, отличный от первоначального размера изображения, требуется дополнительное приведение к общему размеру.

Ниже приведён код, согласно которому происходит потоковое чтение файлов из соответствующих папок исходных размеченных изображений и масок, изменение размера до значений (640,640) и записи в предварительно созданные вложенные папки.

Размер после ресайзинга:

width, height = 640, 640

dsize = (width,height)

Чтение исходного изображения и запись ресайза изображения:

for d, dirs, files in os.walk('./image'):

for f in files:

image = cv2.imread('./image/'+f,cv2.IMREAD\_COLOR)

output = cv2.resize(image,dsize)

#print('./image/'+'resized/'+'resized'+str(f[:-4])+'.jpg')

newname = './image/'+'resized/'+'resized'+str(f[:-4])+'.jpg'

cv2.imwrite(newname,output)

Чтение маски исходного изображения и запись ресайза маски:

for d, dirs, files in os.walk('./mask'):

for f in files:

print(f)

mask = cv2.imread('./mask/'+f,cv2.IMREAD\_COLOR)

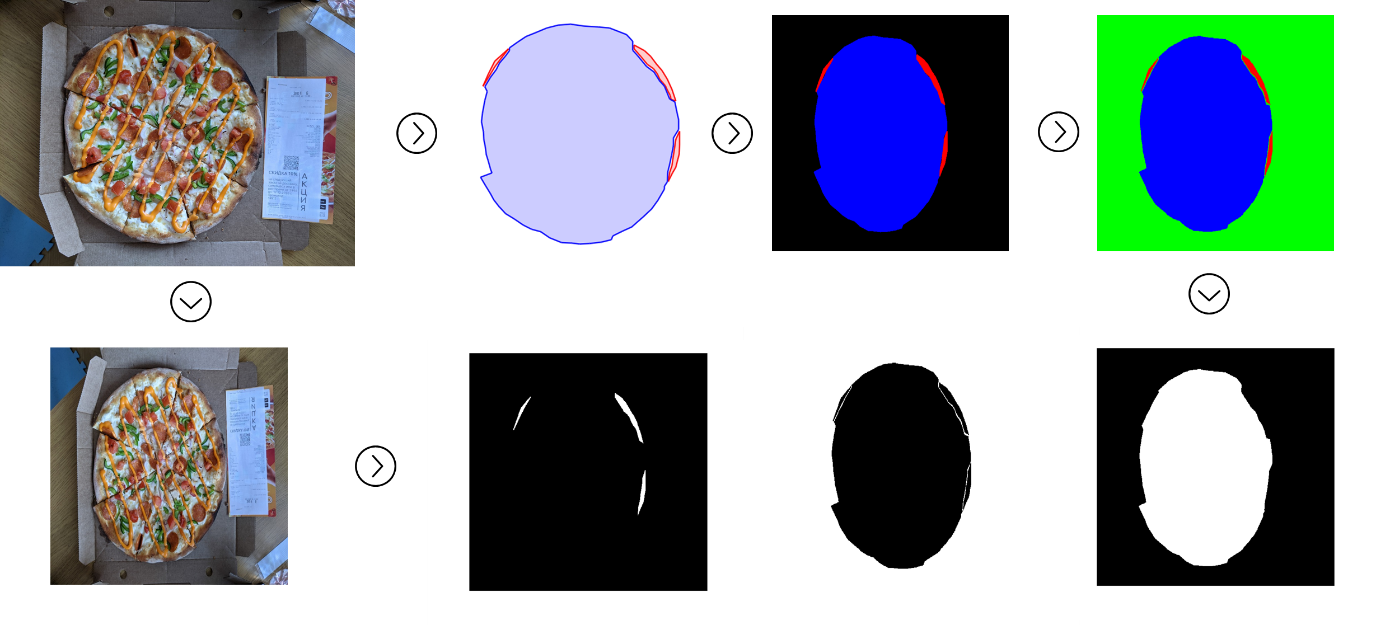
output = cv2.resize(mask,dsize)

newname = './mask/'+'resized/'+'resized'+str(f[:-4])+'.png'

print('./mask/'+'resized/'+'resized'+str(f[:-4])+'.png')

cv2.imwrite(newname,output)

* 1. Визуализация изображения и маски до и после ресайзинга



* 1. Формирование структуры данных

Следующим шагом сформируем список с путём к файлам с нашими данными. Для этого необходимо выполнить два кода.

Для изображений:

data\_dir = '../data/'

image\_lst = []

imres = './image/resized'

for d, dirs, files in os.walk(imres):

for file in files:

fil = data\_dir + imres[2:] + '/' + file

image\_lst.append(fil)  
 print(fil)

Результат выполнения кода для изображений (первые пять строчек):

../data/image/resized/resizedimage\_10066.jpg ../data/image/resized/resizedimage\_10098.jpg ../data/image/resized/resizedimage\_1013.jpg ../data/image/resized/resizedimage\_10171.jpg ../data/image/resized/resizedimage\_10272.jpg

Для масок:

data\_dir = '../data/'

mask\_lst = []

mkres = './mask/resized'

for d, dirs, files in os.walk(mkres):

for file in files:

fil = data\_dir + mkres[2:]+ '/' + file

mask\_lst.append(fil)

print(fil)

Результат выполнения кода для изображений (первые пять строчек):

../data/mask/resized/resizedmask\_10066.png ../data/mask/resized/resizedmask\_10098.png ../data/mask/resized/resizedmask\_1013.png ../data/mask/resized/resizedmask\_10171.png ../data/mask/resized/resizedmask\_10272.png

Таким образом, мы сформировали два списка image\_lst и mask\_lst, в которых находится информация о расположении предварительно приведённых к одному размеру фотографий изображения и соответствующих масок.

* 1. Подготовка данных

Данные были разбиты с коэффициентом 20% на тренировочные и валидационные данные (196 и 49 изображений и 196 и 49 соответствующих изображению масок)

Соответствующая часть кода:

from sklearn.model\_selection import train\_test\_split

img\_train,img\_valid,label\_train,label\_valid = \

train\_test\_split(image\_lst, mask\_lst, test\_size = 0.2, random\_state=5)

В качестве входа на свёрточную сеть U-Net подавалось изображение размером (256,256,3).

Размер батча был равным bs = 3.

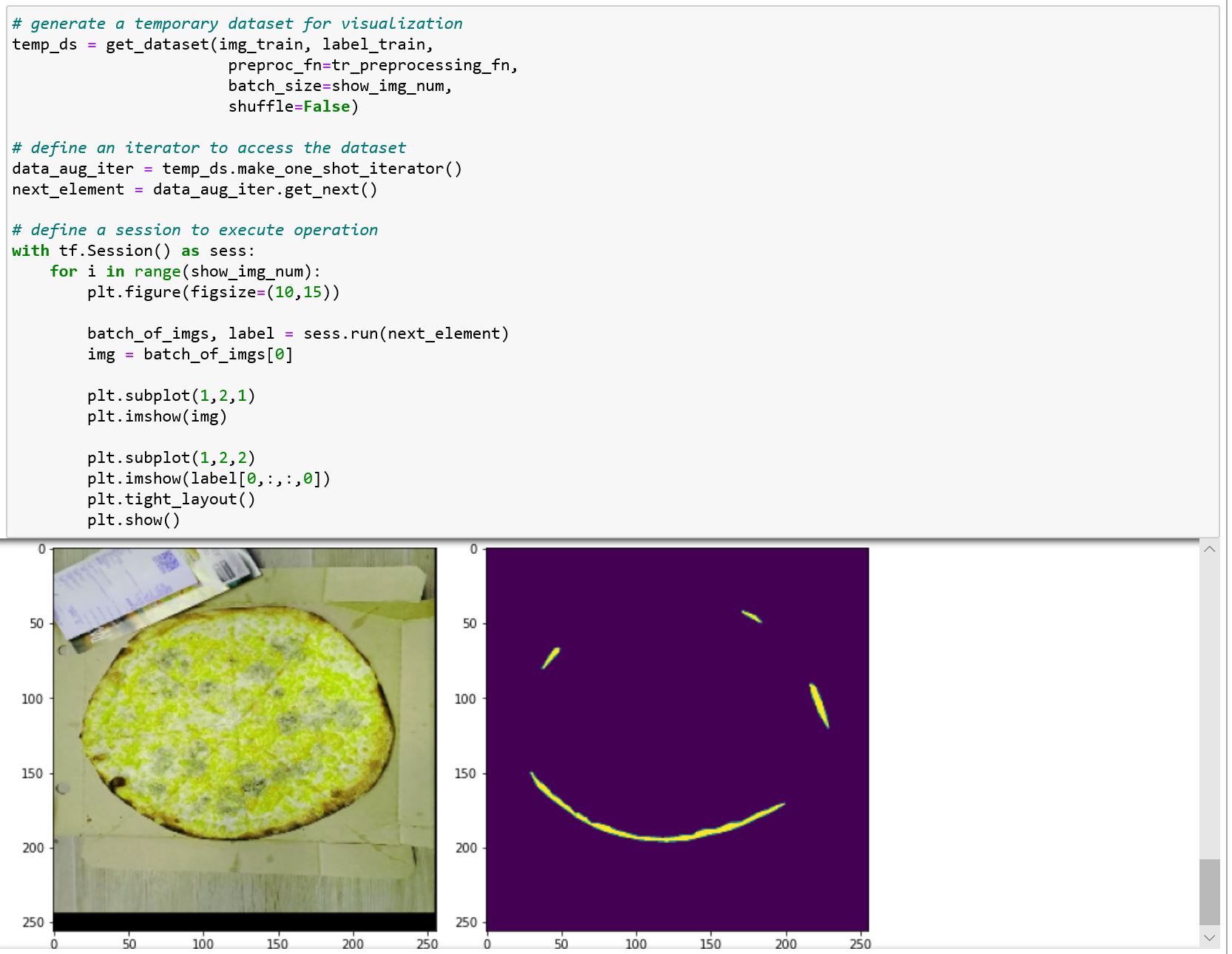
Дополнительно для искусственного увеличения размера датасета использовалась аугментация данных на тренировочных данных: сдвиги, повороты.

Соответствующий код:



* 1. Генерация датасета.

Представим скриншот из программы Jupyter Notebook, на котором показан код и отображение подготовленного изображения и маски.



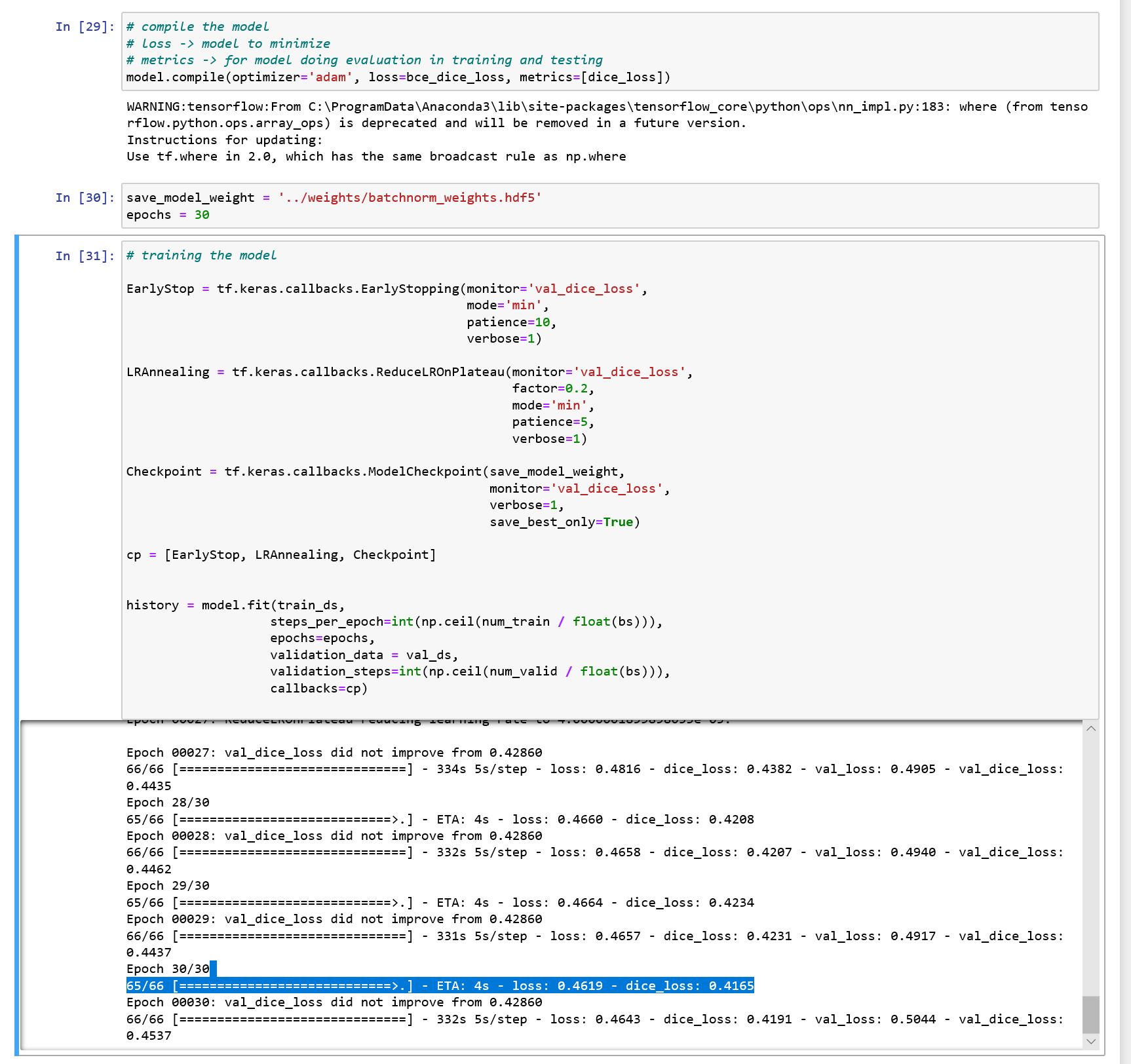
Тем самым, был сформирован датасет, для обучения сети.

* 1. Тренировка модели

Для оценки качества моделей, был установлен параметр количества эпох равным 30. В рамках всех эпох будет происходить 66 итераций на тренировочном и 17 итераций на валидационном датасете, размер батча bs = 3. Для оптимизации сходимости был использован оптимизатор Adam.

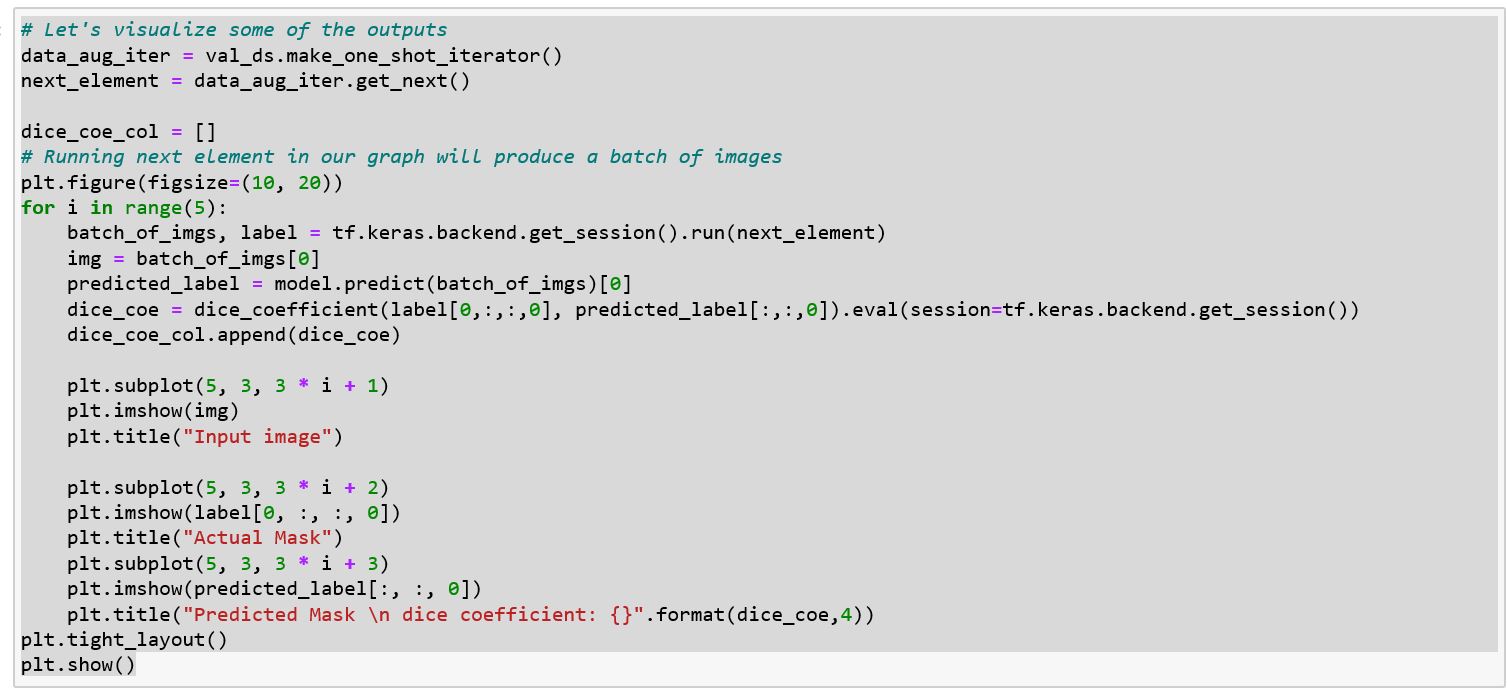
В конце каждой эпохи вызываются функции CALLBACK, которые:  
• пишут в файл веса (сравнивая их с предыдущими и оставляя только лучшие)  
• изменяют коэффициент скорости обучения LR, если скорость обучения маленькая  
• прерывают обучение EarlyStopping, если качество долго не увеличивается

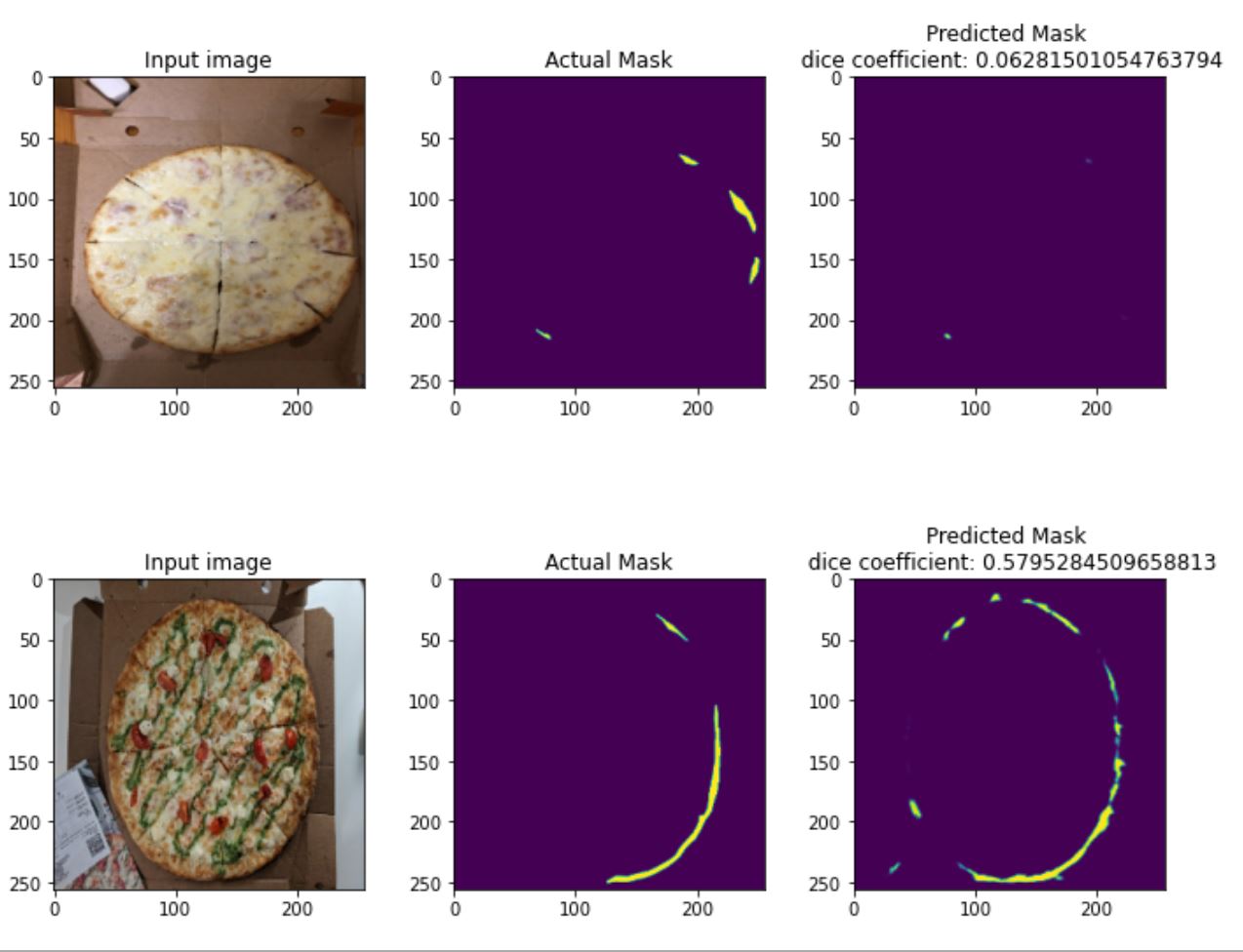
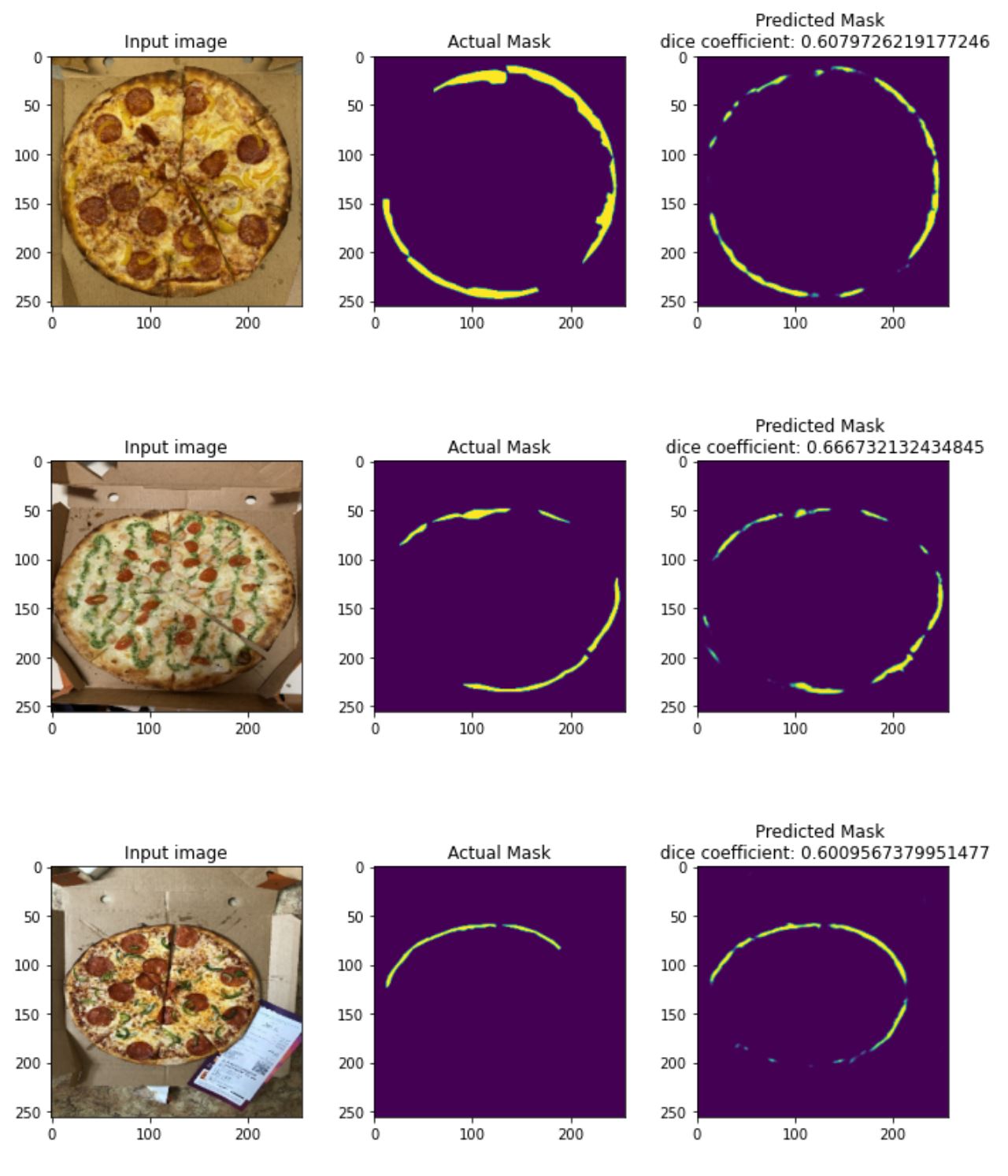
Код:



* 1. Визуализация результатов на валидационной выборке

Следующий код визуализирует результаты обучения модели.



В результате выполнения кода получаем серию изображений:

input image – изображение, взятое из валидационного датасета,

actual mask – маска изображения,

predicted mask – предсказанная сетью маска,

dice coefficient – метрика, коэффициент Дайса.

1. Результаты

При небольшом изначальном количестве материала была произведена аугментация тренировочного датасета и обучена свёрточная сеть U-Net в рамках задачи по семантической сегментации валидационного набора данных, – фотографий пиццы с горелыми краями.

Средний коэффициент Дайса составил avg\_dice\_coe = 0.50

Максимальное значение составило 0.67

Минимальное значение составило 0.06

1. Выводы и заключения

В рамках дипломного проекта был выбран кейс DODO по обнаружению дефекта на пицце (горелых бортов) и проведена семантическая сегментация горелых бортов. Результат вполне приемлимый, исходя из количества доступных фотографий в первоначальном датасете и выбора в качестве алгоритма для обнаружения горелых бортов свёрточной сети U-Net.

В качестве улучшения результатов моделирования следует увеличить количество фотографий с дефектом – ‘горелый борт’.

Следующим шагом следует более взвешенно отнестись к разметке изображений: не допускать к разметке изображения с неявным горелым бортом, не особо чёткие фотографии горелых бортов, неявные границы между горелым бортом и фоном.

1. Список источников

1. [Метрика Дайса](https://ru.qaz.wiki/wiki/S%C3%B8rensen%E2%80%93Dice_coefficient)

2. Статья авторов U-Net: [U-Net: Convolutional Networks for Biomedical Image Segmentation](https://arxiv.org/pdf/1505.04597.pdf)

3. [Open Source Data Labeling Tool](https://labelstud.io/) (https://labelstud.io/)