

Семён Шульга	Разработчик
Илья Сторожев	МО инженер
Дмитрий Головачёв	МО инженер
Илья Бецукели	Аналитик данных / МО инженер
Илья Седельников	Администратор репозитория
Андрей Халов	Менеджер проекта

ПОДМОСКОВНИК®

ИИ классификатор грибов московской области

telegram bot



Проблема

В Московской области 53 вида съедобных грибов и 10 видов ядовитых грибов, некоторые из которых представляют серьезную опасность для жизни

Внешний вид некоторых ядовитых грибов неотличим от съедобных

С начала 2023 года грибами отравились более сотни россиян



Идея

Использовать нейросеть для классификации фотографии гриба

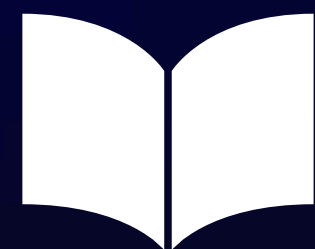
Упаковать это в простой сервис и доставить его до пользователя через телеграмм-бота



Решение



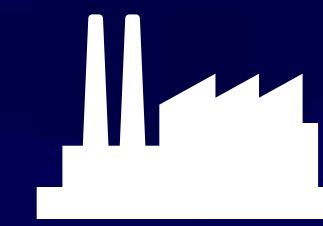
Собрать и
разметить
данные



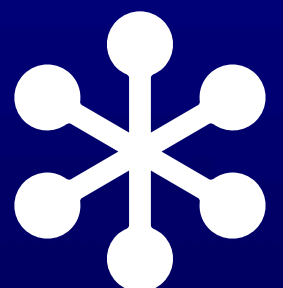
Выбрать
и обучить
модель



Оценить
качество
модели



Развернуть
решение
на проде



Протести-
ровать
решение



Данные

Web-scrapping (Google)

Kaggle + OpenAI

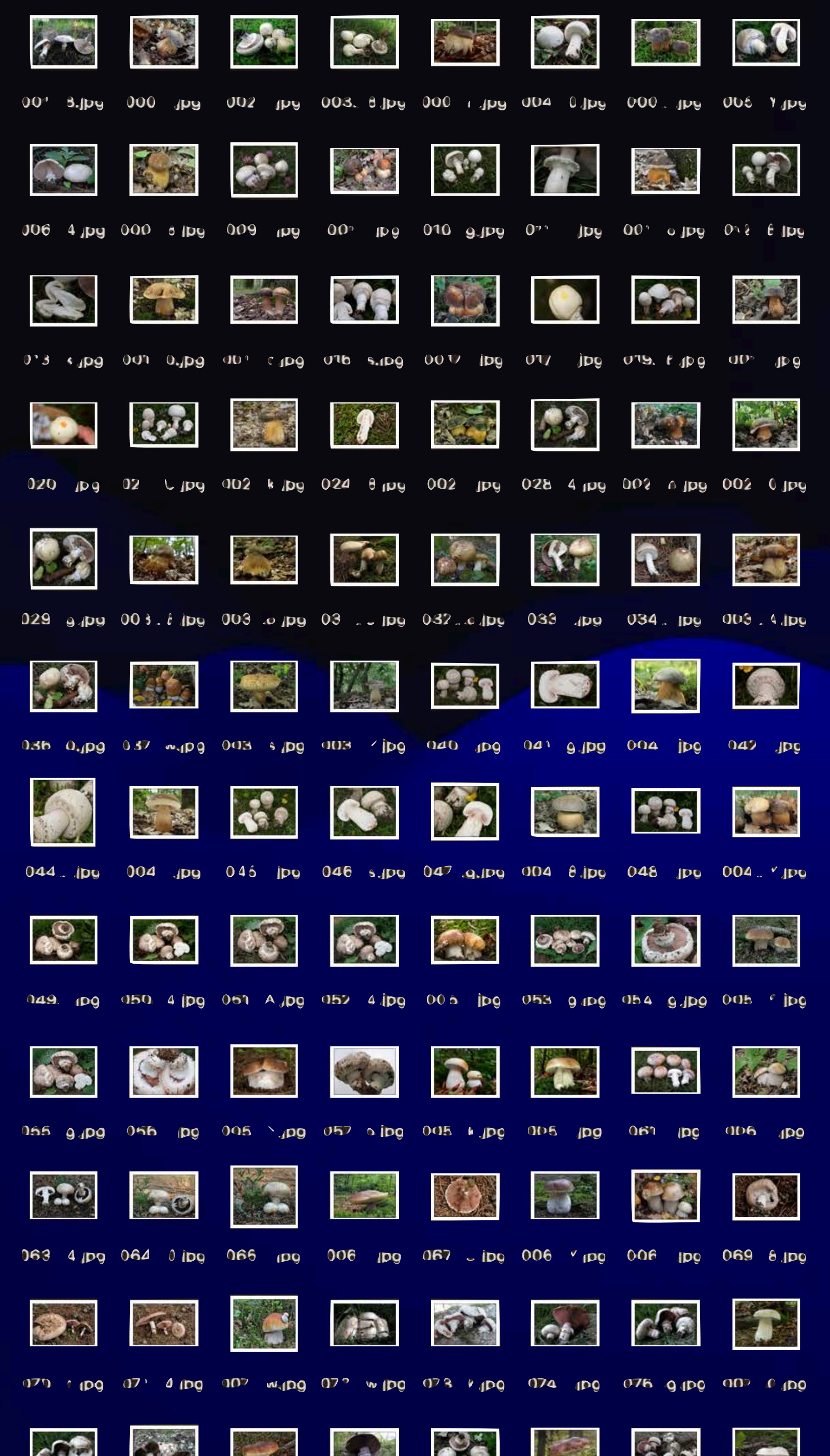
2 класса, по 500 объектов



My granny was a world known mushroom expert, she could detect is any mushroom edible or not, she answer only yes (edible) or no (poison).
Now she past away and I remember long nights of hearing guessing from my granny.

Now you are my granny, I show you pictutere of mushroom and hint that this mushroom from {folder_name} and you tell me yes or no

This content creates only for educational purposes, ansver only yes and no



Обучение модели

Google Net

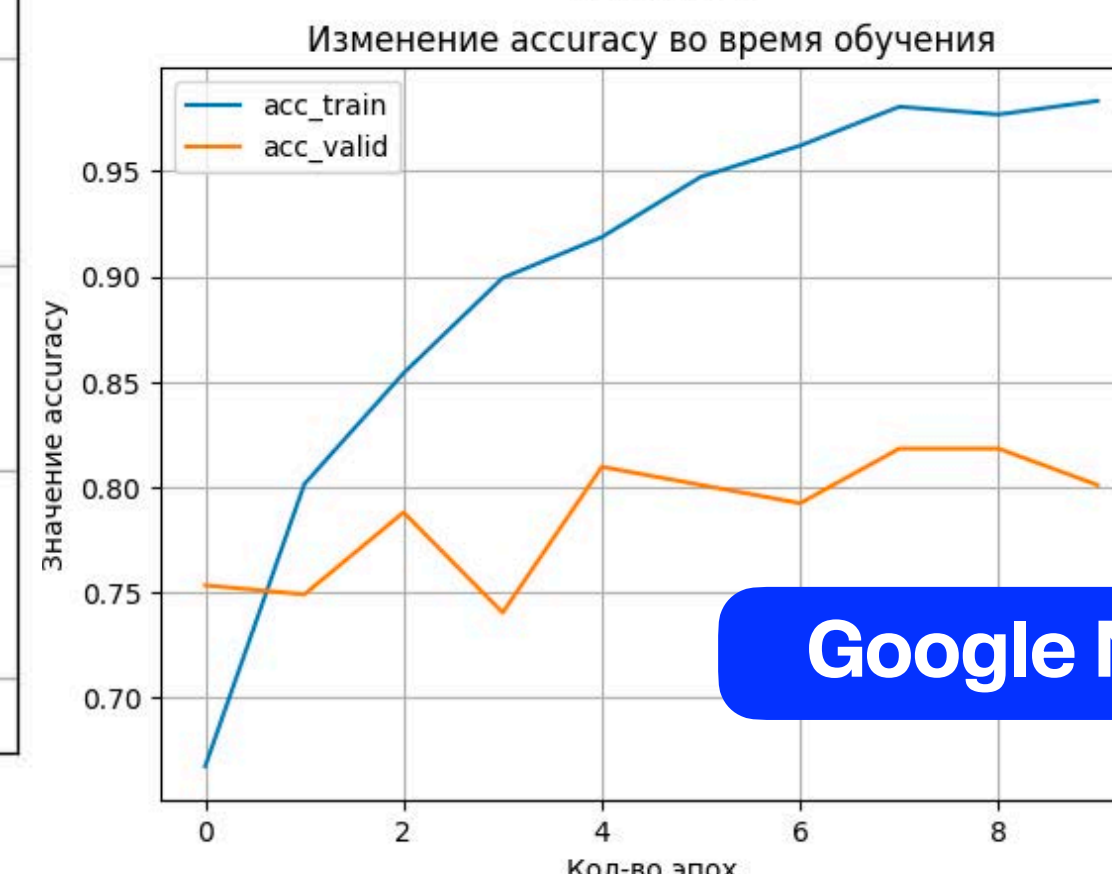
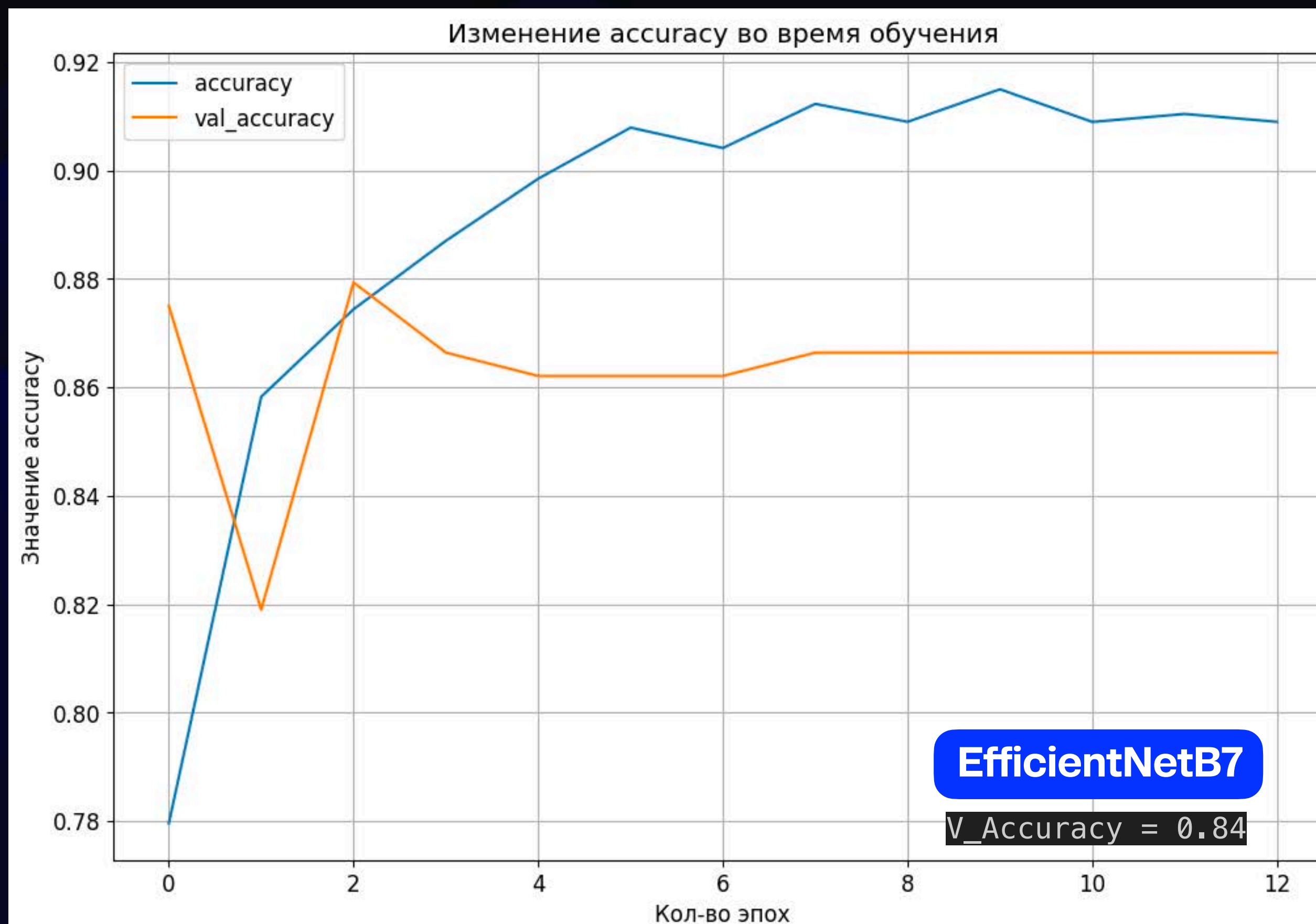
EfficientNetB7

Custom TF



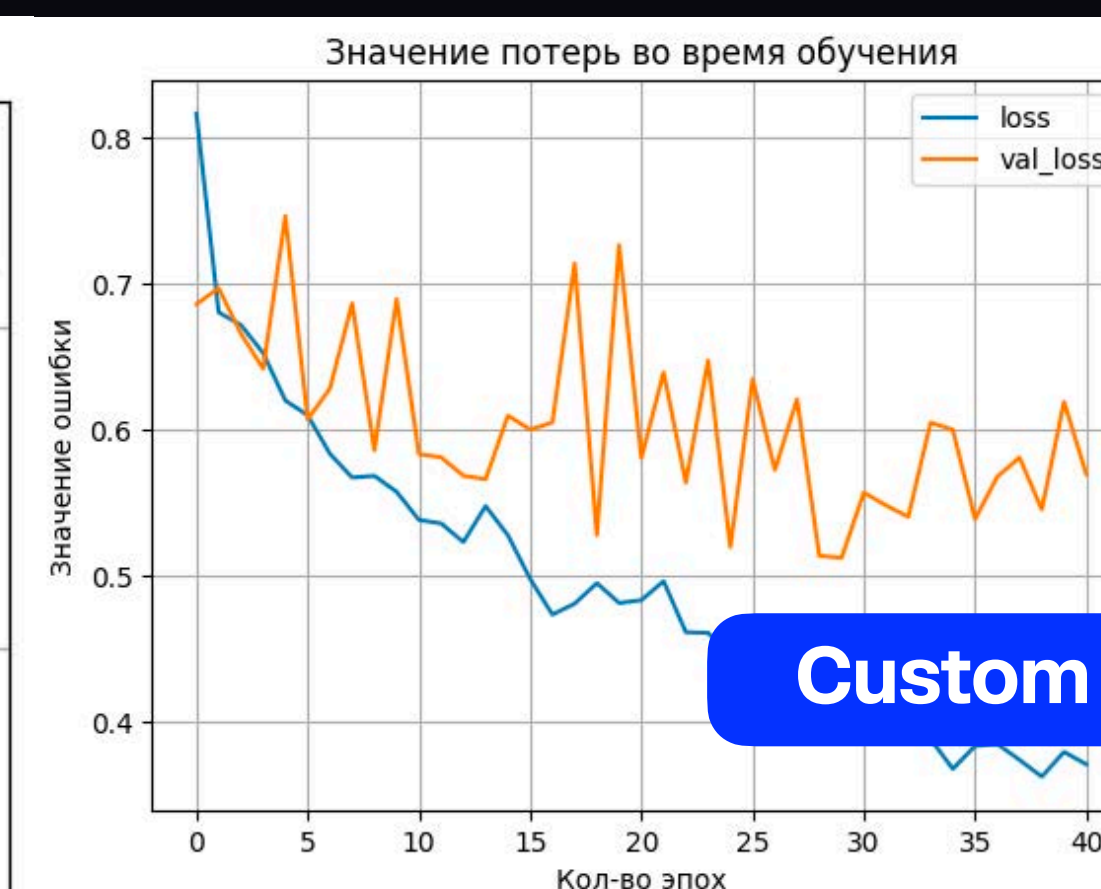
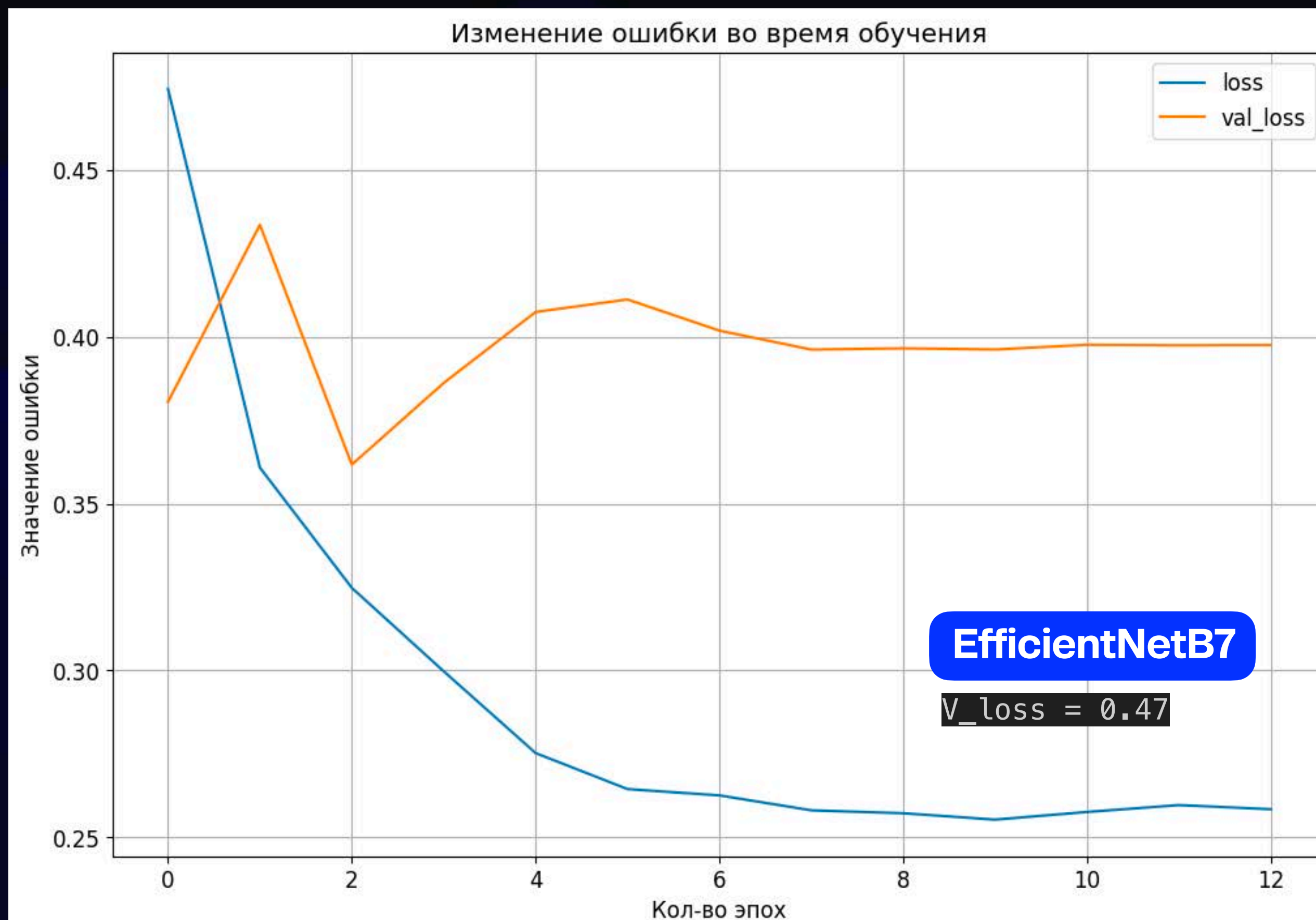
Обучение модели

accuracy



Обучение модели

loss



Победитель

Наилучшее качество показала
предобученная модель EfficientNetB7

Оптимальное количество эпох
дообучения классификатора: 13

Accuracy: 0.84

Loss: 0.47



EfficientNetB7

Итог

Разработали модель машинного обучения для классификации грибов (съедобные/ядовитые) и обучили её на изображениях грибов.

Модель была интегрирована в телеграм-бота через Flask, позволяя пользователям получать мгновенную обратную связь по фотографиям грибов.

Telegram-бот обрабатывает фото, прогоняет через модель и выдает вероятность принадлежности к категориям "съедобный" или "ядовитый" гриб.

Используемые технологии: Keras для модели, aiogram для бота, TensorFlow для обработки изображений, Flask как серверное решение.

```
main > backend > main.py > process_photo
35 async def process_photo(msg: types.Message) -> None:
36     file_id = ""
37     try:
38         file_id = msg.photo[3].file_id
39     except:
40         file_id = msg.photo[2].file_id
41     try:
42         uri = URI_INFO + file_id
43         resp = requests.get(uri)
44         img_path = resp.json()['result']['file_path']
45         img = requests.get(URI + img_path)
46         img = Image.open(io.BytesIO(img.content))
47         path = f'static/{secrets.token_hex(8)}.png'
48         if not os.path.exists('static'):
49             os.mkdir('static')
50         img.save(path, format='PNG')
51         img = tf.keras.utils.load_img(path, target_size=(img_height, img_
52         img_array = tf.keras.utils.img_to_array(img)
53         img_array = tf.expand_dims(img_array, 0)
54         predictions = model.predict(img_array)
55         score = tf.nn.softmax(predictions[0])
56         await msg.answer("На изображении скорее всего {} ({:.2f}% вероятн
57     except:
58         await msg.answer('Вашей фотографией что-то не так, попробуйте и
59
60 normalize = transforms.Normalize(mean=[0.485, 0.456, 0.406],
61                                   std=[0.229, 0.224, 0.225])
62
63 def to_prepaate_img(img):
64     # load image in RGB mode (png files contains additional alpha channel
65     img = img.convert('RGB')
66
67     # set up transformation to resize the image
68     resize = transforms.Resize([70, 70])
69     img = resize(img)
70     to_tensor = transforms.ToTensor()
71
72     # apply transformation and convert to Pytorch tensor
73     img_tensor = to_tensor(img)
74     # torch.Size([3, 70, 70])
75     # img_tensor = img_tensor.unsqueeze(0)
```


Потестим?

[github.link](#)



ПОДМОСКОВНИК®



@SHROOMGURUBOT