

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**



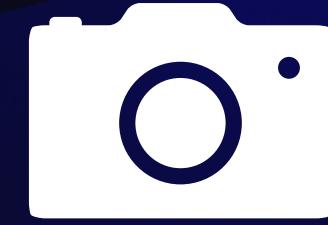
# Идея

Использовать нейросеть для классификации фотографии гриба

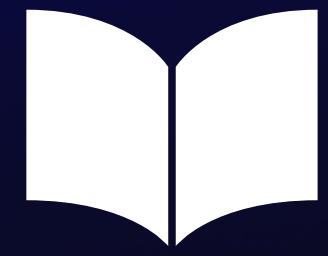
Упаковать это в простой сервис и доставить его до пользователя через телеграмм-бота



# Решение



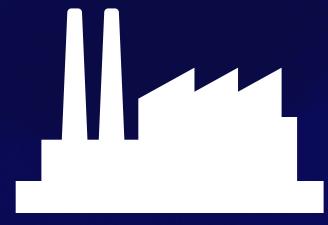
Собрать и  
разметить  
данные



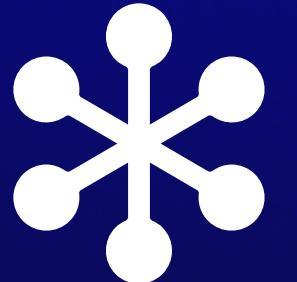
Выбрать  
и обучить  
модель



Оценить  
качество  
модели



Развернуть  
решение  
на проде



Протести-  
ровать  
решение

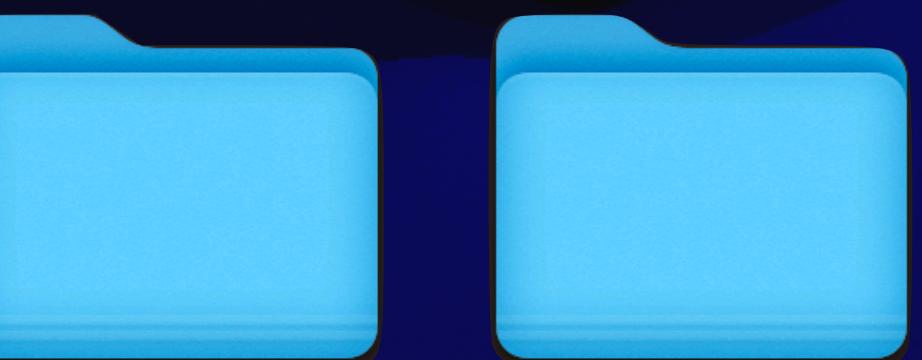


# данные

Web-scraping (Google)

Kaggle + OpenAI

2 класса, по 500 объектов



don't\_eat\_me

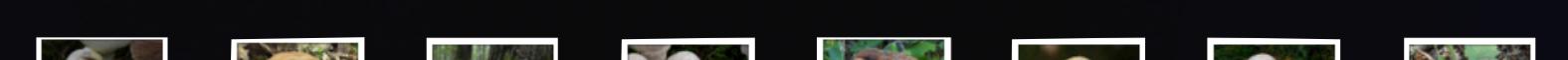
eat\_me



001\_8.jpg 000\_1.jpg 002\_1.jpg 003\_8.jpg 000\_1.jpg 004\_0.jpg 000\_1.jpg 005\_1.jpg



006\_4.jpg 000\_5.jpg 009\_1.jpg 007\_1.jpg 010\_9.jpg 009\_1.jpg 008\_0.jpg 012\_6.jpg



013\_4.jpg 001\_0.jpg 001\_3.jpg 016\_8.jpg 0017\_1.jpg 017\_1.jpg 019\_6.jpg 019\_1.jpg



020\_1.jpg 021\_1.jpg 002\_4.jpg 024\_8.jpg 002\_1.jpg 028\_4.jpg 002\_0.jpg 002\_0.jpg



029\_9.jpg 003\_6.jpg 003\_0.jpg 03\_1\_1.jpg 032\_0\_0.jpg 033\_4.jpg 034\_1.jpg 003\_4.jpg



036\_0.jpg 037\_0.jpg 003\_6.jpg 003\_1.jpg 046\_1.jpg 041\_9.jpg 004\_1.jpg 042\_1.jpg



044\_1.jpg 004\_1.jpg 045\_1.jpg 046\_8.jpg 047\_9.jpg 004\_8.jpg 048\_1.jpg 004\_7.jpg



049\_1.jpg 050\_4.jpg 051\_A.jpg 052\_4.jpg 005\_1.jpg 053\_9.jpg 054\_9.jpg 005\_6.jpg



056\_9.jpg 056\_1.jpg 005\_1.jpg 057\_0.jpg 005\_6.jpg 006\_1.jpg 061\_1.jpg 006\_1.jpg



063\_4.jpg 064\_0.jpg 066\_1.jpg 006\_1.jpg 067\_1.jpg 006\_V.jpg 006\_1.jpg 069\_8.jpg



070\_1.jpg 071\_4.jpg 072\_W.jpg 072\_W.jpg 073\_V.jpg 074\_1.jpg 075\_9.jpg 077\_0.jpg



My granny was a world known mushroom expert, she could detect if 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 picture of mushroom and hint that this mushroom from {folder\_name} and you tell me yes or no

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

# Обучение модели

Google Net

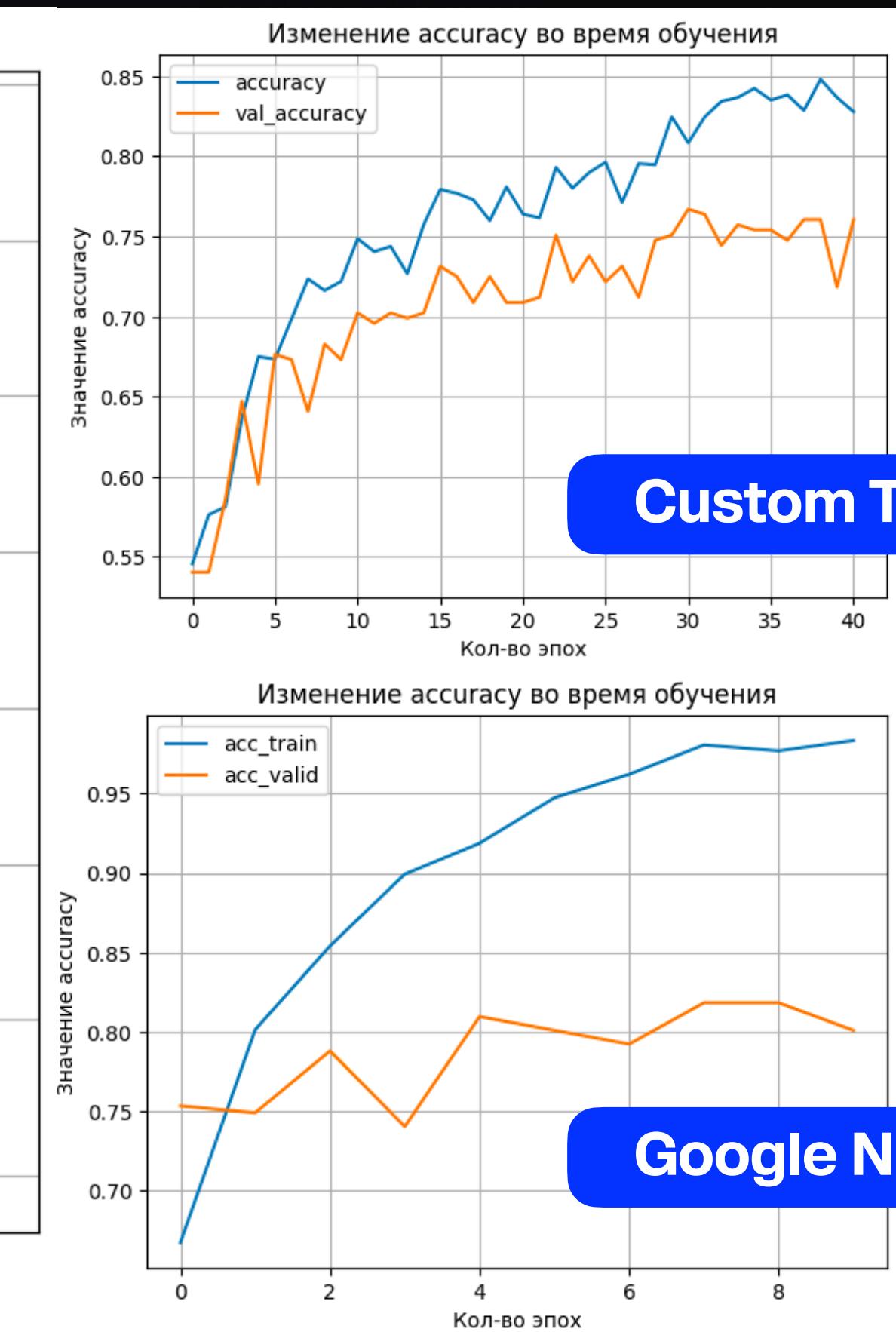
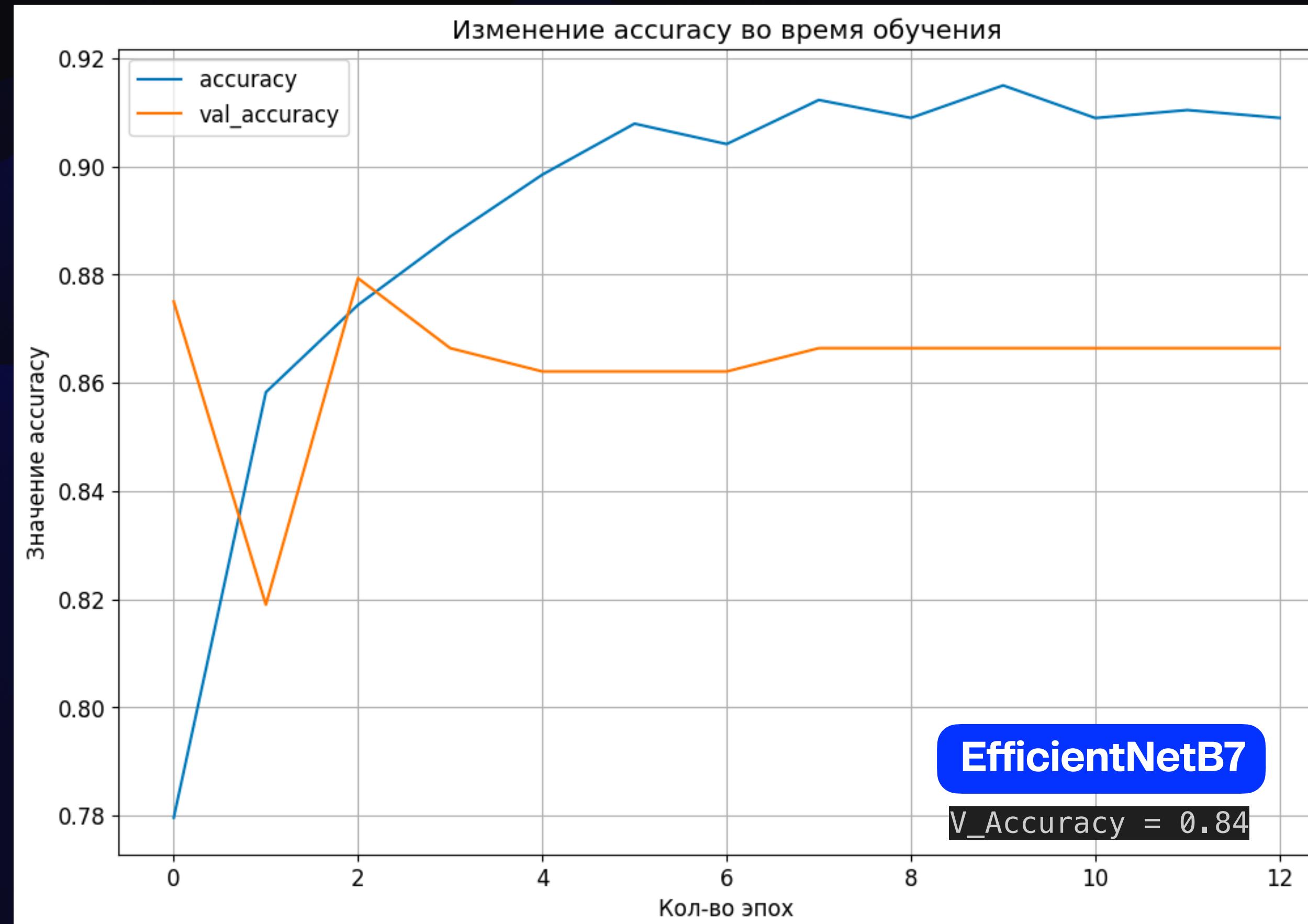
Custom TF

EfficientNetB7



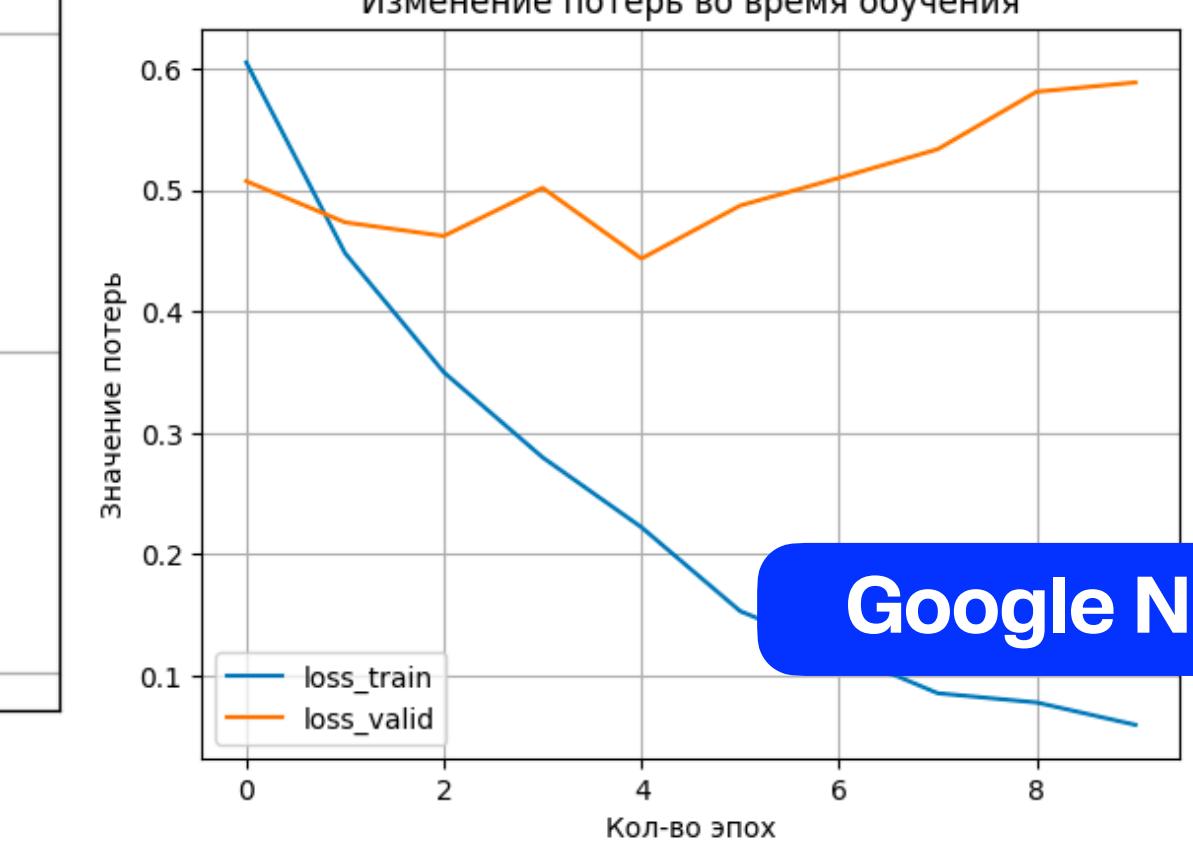
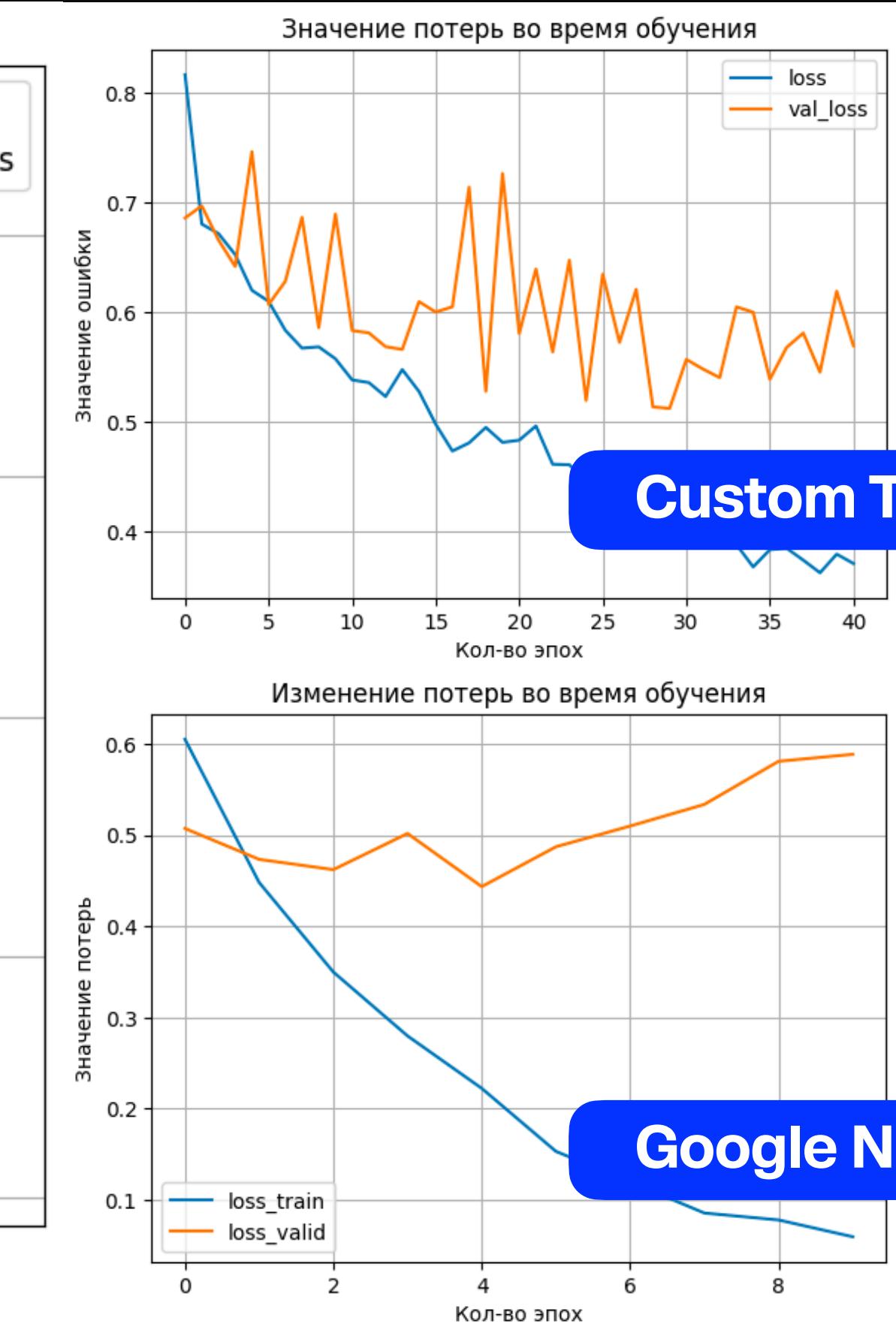
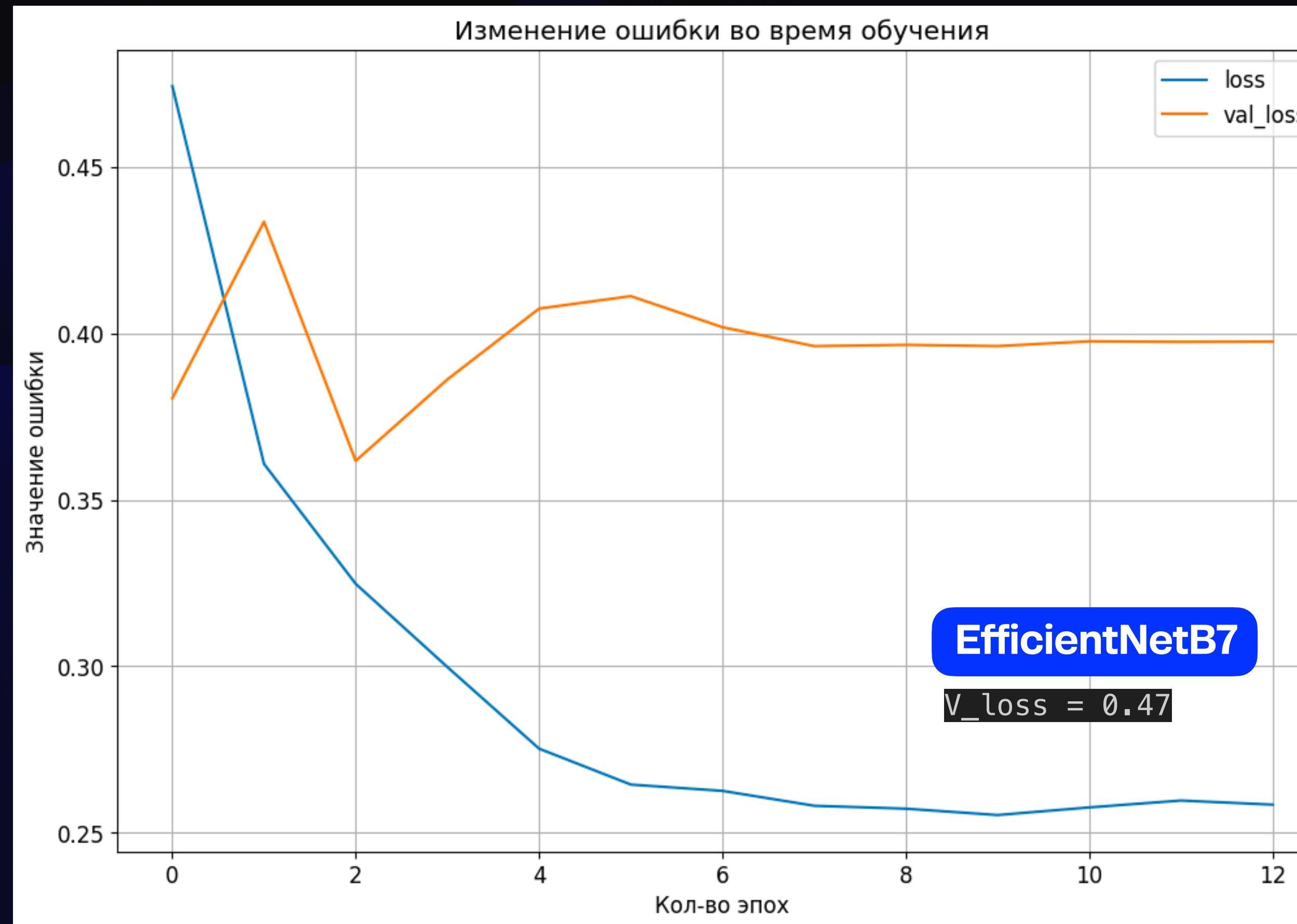
# Обучение модели

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.

Телеграм-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
38         try:
39             file_id = msg.photo[3].file_id
40         except:
41             file_id = msg.photo[2].file_id
42
43         try:
44             uri = URI_INFO + file_id
45             resp = requests.get(uri)
46             img_path = resp.json()['result']['file_path']
47             img = requests.get(URI + img_path)
48             img = Image.open(io.BytesIO(img.content))
49             path = f'static/{secrets.token_hex(8)}.png'
50             if not os.path.exists('static'):
51                 os.mkdir('static')
52             img.save(path, format='PNG')
53             img_array = tf.keras.utils.load_img(path, target_size=(img_height, img_
54             img_array = tf.keras.utils.img_to_array(img)
55             img_array = tf.expand_dims(img_array, 0)
56             predictions = model.predict(img_array)
57             score = tf.nn.softmax(predictions[0])
58             await msg.answer("На изображении скорее всего {} {:.2f}% вероятн
59
60         except:
61             await msg.answer('С вашей фотографией что-то не так, попробуйте и
62
63     def to_prepaaate_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_tenzer = to_tensor(img)
74         # torch.Size([3, 70, 70])
75
76         return img_tenzer
```

# Потестим?

[github.link](#)



ПОДМОСКОВНИК®

МФТИ



@SHROOMGURUBOT

