



UNIVERSITY
OF ARTIFICIAL
INTELLIGENCE

Соколов Алексей Александрович

Дизайн маникюра на фотографии

Дата защиты: 07.05.2022

Дипломный
руководитель
Никита Серов
Поток обучения
Февраль, 2021



Постановка задачи

- На фотографии кистей рук пользователя необходимо распознать области ногтей
- Получить маску изображения классифицированную по двум классам: «ногти» и «остальное»
- По маске перекрасить области ногтей в цвет выбранный пользователем

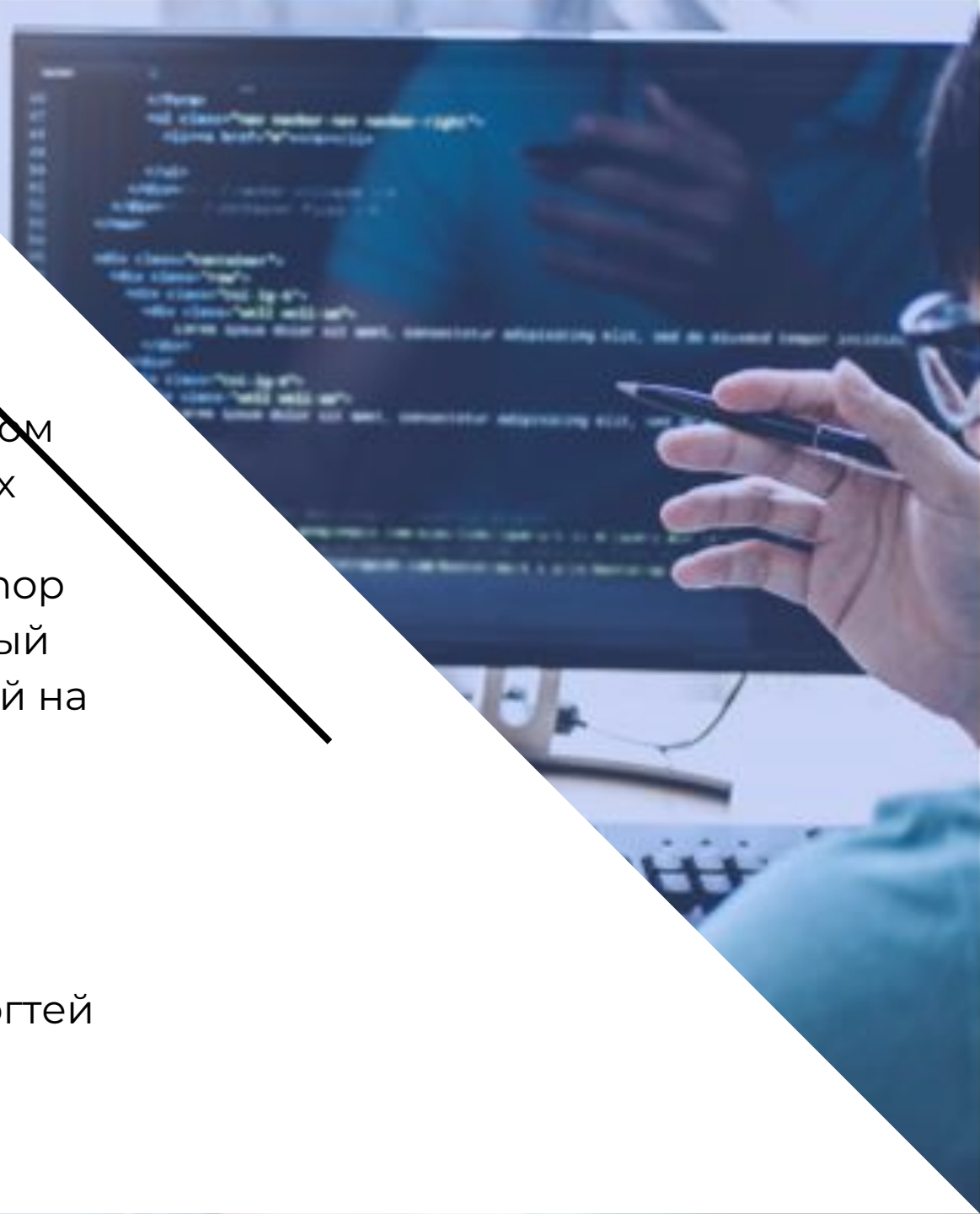


Цель

- Создание WEB-приложения
«Дизайн маникюра с
помощью ИИ»

Задачи

- Собрать обучающую базу фото рук с маникюром посредством парсинга на тематических сайтах
- На фото собранной базы в программе Photoshop закрасить области ногтей в один определённый цвет для обучения НС находить области ногтей на фото
- Написать архитектуру НС и обучить её распознавать области ногтей на фото
- Написать скрипт перекрашивания области ногтей в требуемый цвет



Обучающая база

Объем базы

Парсингом тематического сайта получено 5077 фото
дизайна маникюра

Источник сбора данных

<https://bestmanikyur.ru/>

Инструменты для сбора данных

IDE PyCharm и библиотека «BeautifulSoup» для
парсинга HTML и XML документов



Популярный дизайн ногтей и образцы маникюра 2021



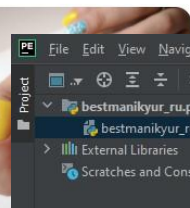
Маникюр №5077

#Воздушный маникюр
#Идеи разноцветного маникюра
#Маникюр в летний отпуск
#Маникюр лето 2021
#Маникюр на день рождения
#Маникюр на июль
#Маникюр под летнее платье
#Многоцветный маникюр



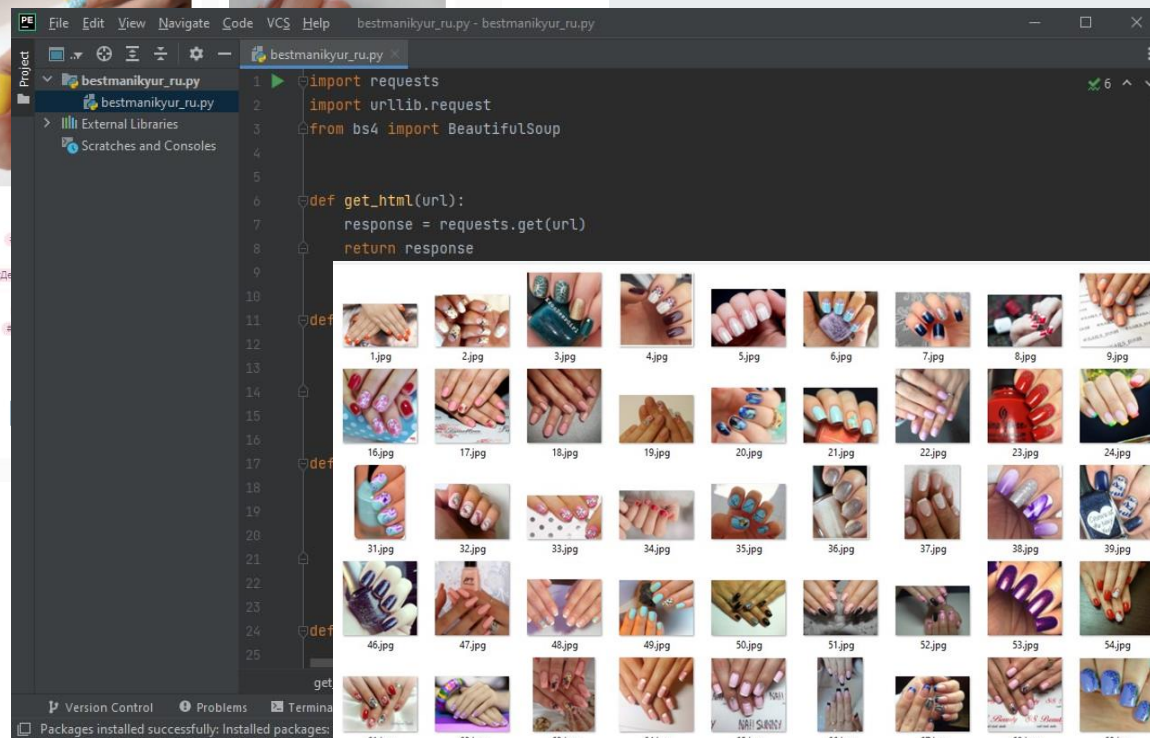
Маникюр №5076

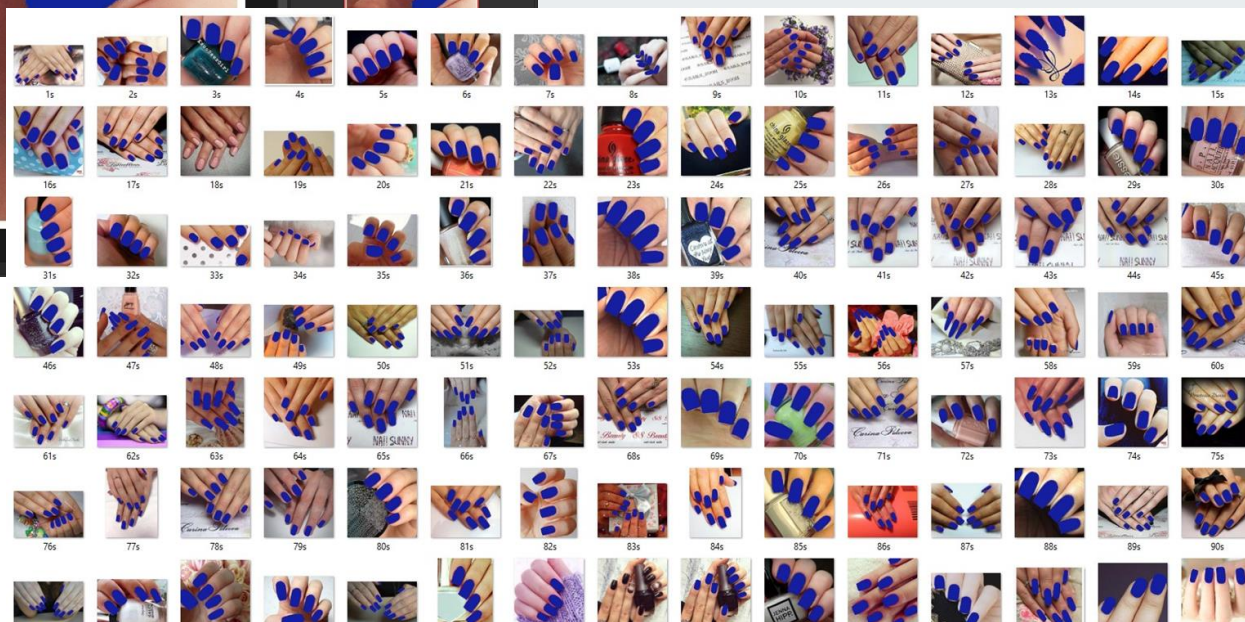
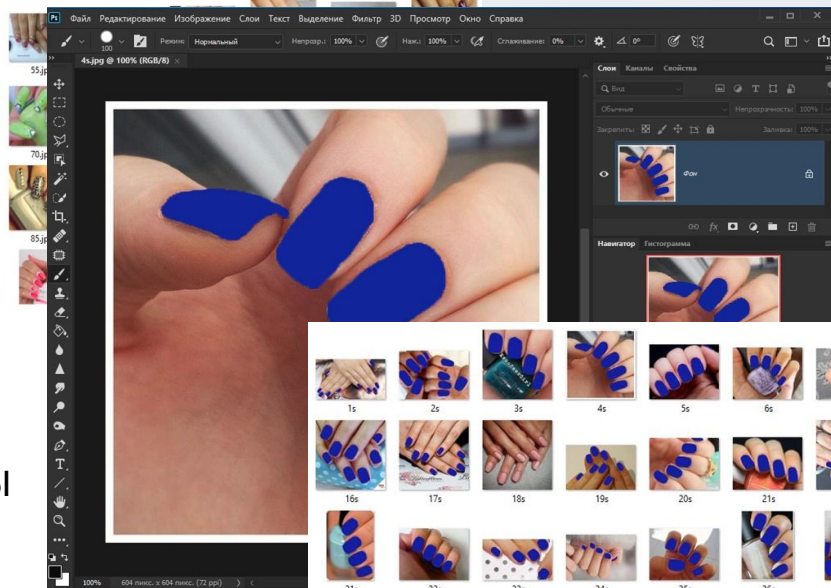
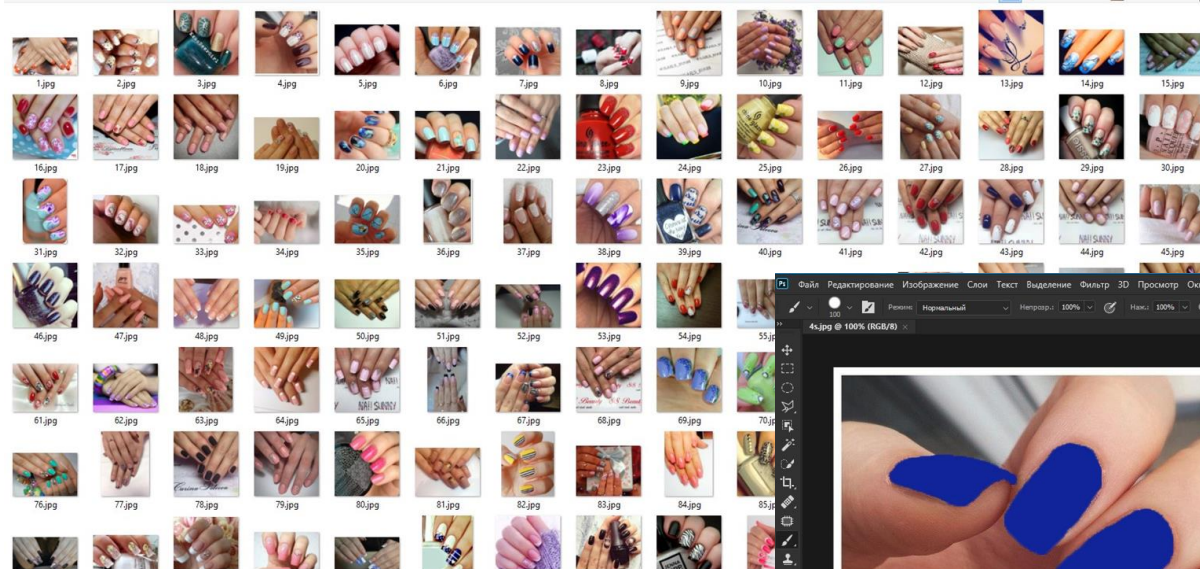
#Интересный маникюр
#Красивый маникюр
#Маникюр на вечеринку
#Маникюр с абстрактным рисунком
#Модные тенденции маникюра 2020
#Необычный маникюр
#Неординарный маникюр
#Новые идеи маникюра



Маникюр №5075

#Дизайн ногтей

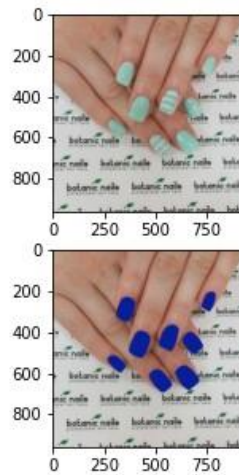
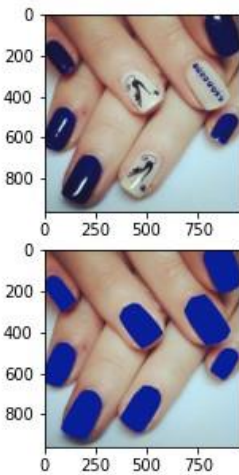
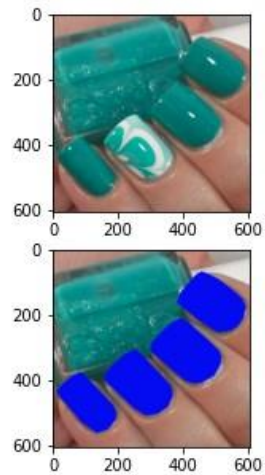




На 400+ фото собранной базы
в программе Photoshop были
закрашены области ногтей в
один определённый цвет для
обучения ИС находить
данные области

Обучающая база

- В первую очередь необходимо было определить минимальные размеры изображений в датасете, для подачи на обучение сети в едином размере



Минимальный размер изображение по x: 339
Минимальный размер изображение по y: 316

Обучающая база

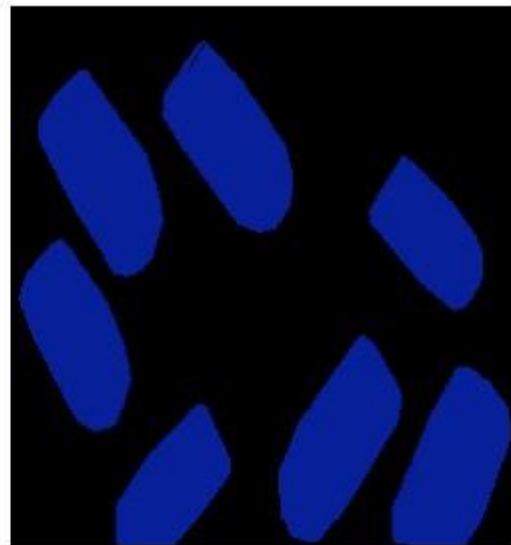
- Так как в архитектуре сети будет несколько слоёв MaxPooling2D необходимо задать размеры в чётных значениях, достаточных для целочисленного деления на 2 столько раз, сколько будет данных слоёв

```
1 # Задаём минимальные значения по x и y для обучения модели
2 img_width = 272
3 img_height = 256
4 num_classes = 2
```



Обучающая база

- Из изображений формируем X-train и Y-train в виде numpy-массивов
- Y-train переводим в формат One Hot Encoding
- X-train и Y-train разделяем на обучающую и проверочную выборки

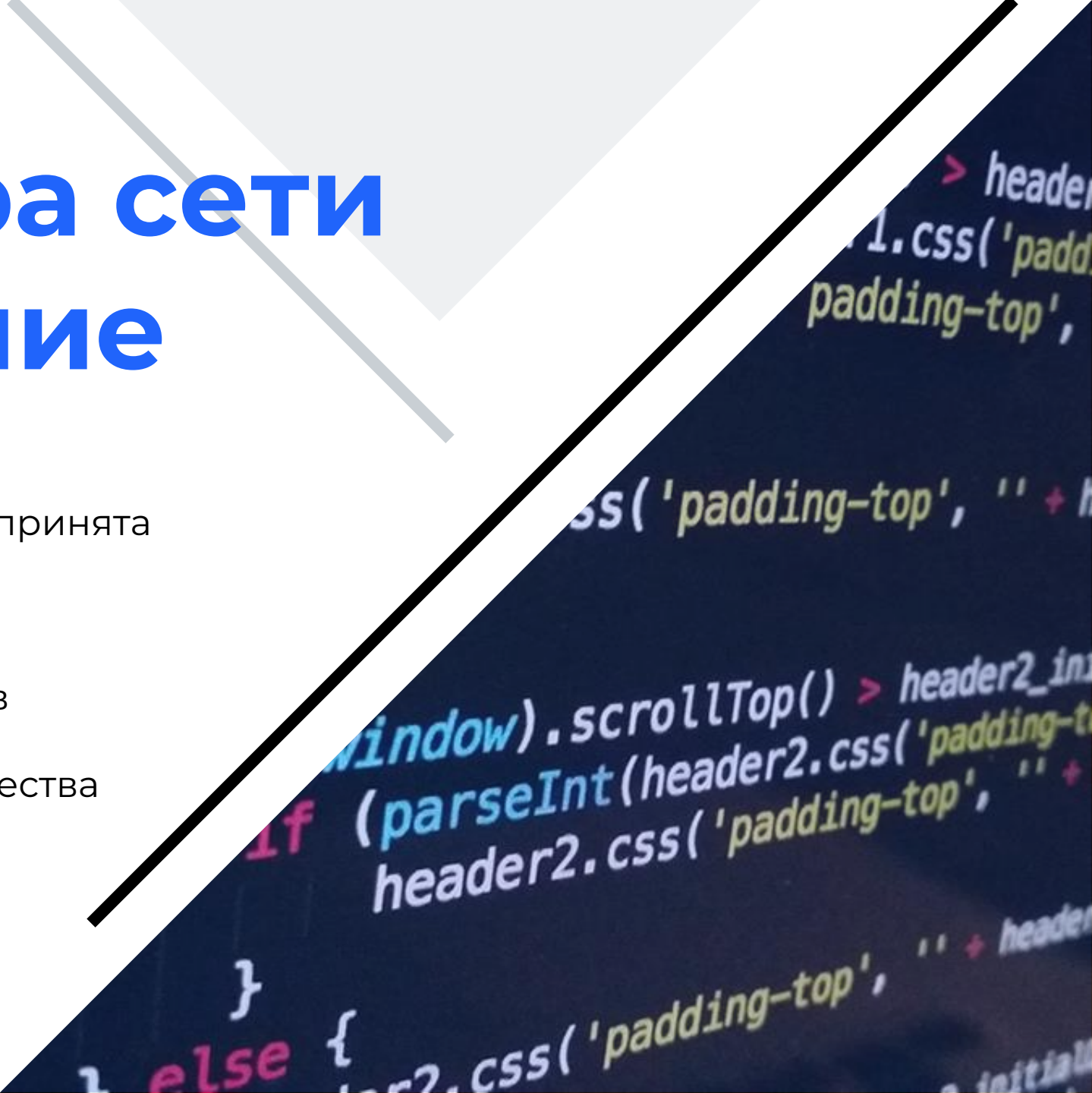


Архитектура сети и её обучение

В качестве основной гипотезы была принята U-Net-архитектура сети

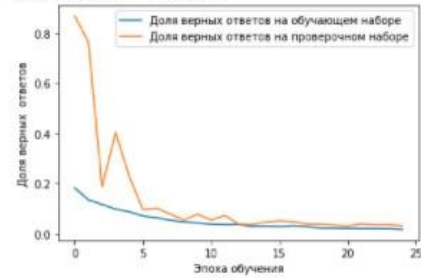
Для U-Net характерно:

- достижение высоких результатов в различных реальных задачах
- использование небольшого количества данных для достижения хороших результатов



Стартовое обучение: 25 эпох, batch_size=8,
activation='softmax', loss='categorical_crossentropy'

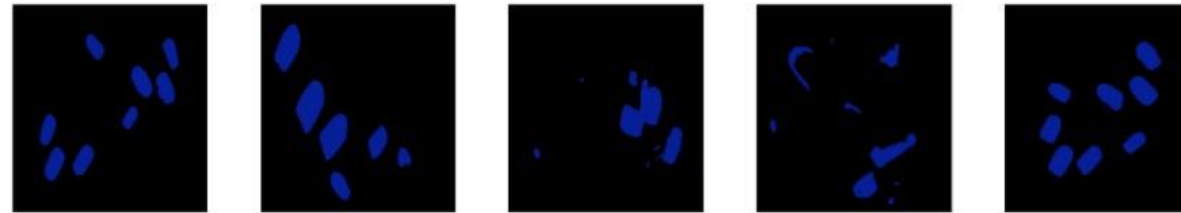
50/50 [=====] - 94s 25/step - loss: 0.0284 - dice_loss: 0.0163 - val_loss: 0.0910 - val_dice_loss: 0.0306
Время обучения: 2369.9 с



val_dice_loss: 0.0306

```
1 #Распознавание  
2 processImage(bs8_relu_ep25_model_d1, 2, 5)
```

Predict



Сегментация

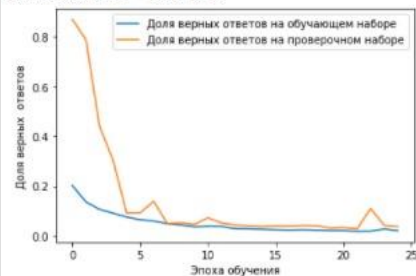


Исходное
изображение



Гипотеза: 25 эпох, batch_size=8,
activation='sigmoid', loss='binary_crossentropy'

50/50 [=====] - 86s 2s/step - loss: 0.0353 - dice_loss: 0.0206 - val_loss: 0.1012 - val_dice_loss: 0.0364
Время обучения: 2188.91 c



val_dice_loss: 0.0364

1 #Распознавание
2 processImage(bs8_relu_ep25_model_d1_bc_sigm, 2, 5)

Predict



Сегментация

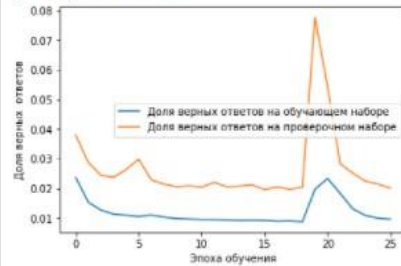


Исходное изображение



Гипотеза: U-Net, **50** эпох, batch_size=8, с **callback'ом** (фиксация лучшей эпохи и остановкой обучения если **val_dice_loss** не улучшается более 10 эпох)

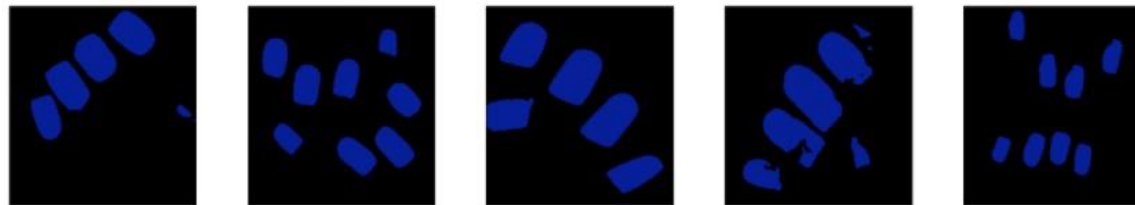
```
Epoch 26/50  
50/50 [=====] - ETA: 0s - loss: 0.0159 - dice_loss: 0.0095Restoring model weights from the end of the best epoch.  
50/50 [=====] - 39s 771ms/step - loss: 0.0159 - dice_loss: 0.0095 - val_loss: 0.0578 - val_dice_loss: 0.0200  
Epoch 00026: early stopping  
Epoch with best val_dice_loss: 00016  
val_dice_loss: 0.019603043794631958  
Время обучения: 991.38 с
```



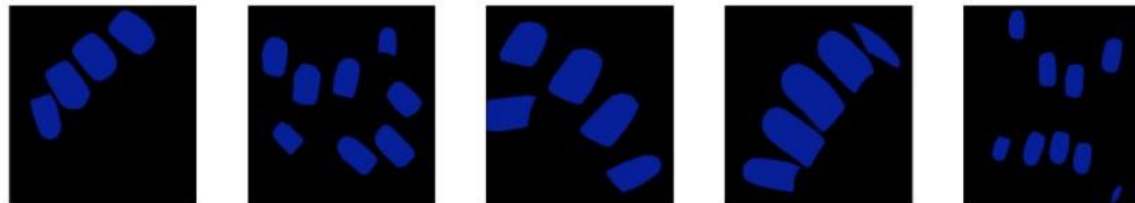
val_dice_loss: 0.0196

```
1 #Распознавание  
2 processImage(bs8_relu_ep50_model_fin, 2, 5)
```

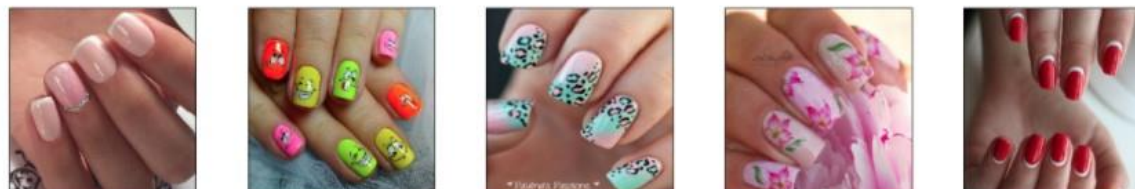
Predict



Сегментация



Исходное
изображение



Аугментация

Расширена обучающая и проверочная база до 1300+ изображений



Исходное изображение



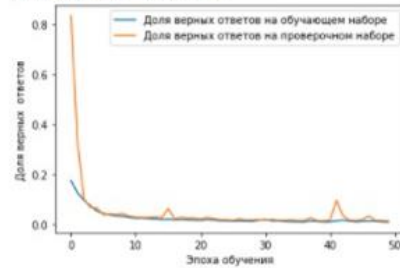
Зеркальное отражение по горизонтали



Зеркальное отражение по вертикали

Гипотеза: U-Net, 50 эпох, с callback'ом и аугментацией

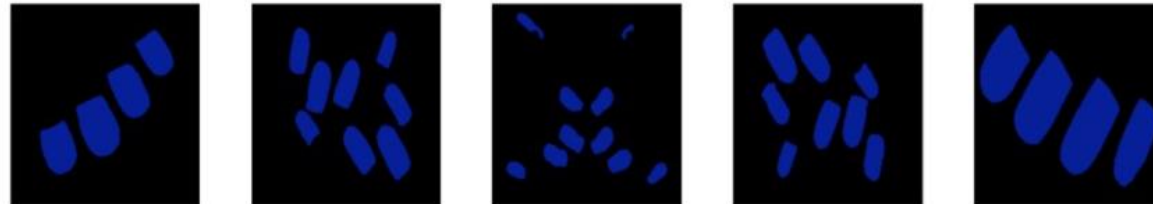
Epoch 50/50
150/150 [=====] - 260s 2s/step - loss: 0.0157 - dice_loss: 0.0095 - val_loss: 0.0355 - val_dice_loss: 0.0135
Epoch with best val_dice_loss: 00050
val_dice_loss: 0.013517155312001705
Время обучения: 13053.36 c



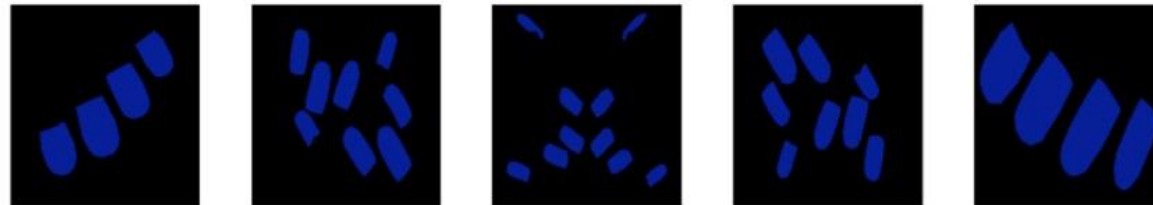
val_dice_loss: 0.0135

```
1 # Распознавание  
2 processImage(bs8_relu_ep50_model_aug_fin, 2, 5)
```

Predict



Сегментация

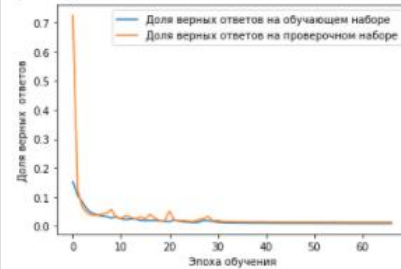


Исходное изображение



Гипотеза: U-Net, **100** эпох с callback'ом+ (если val_dice_loss не улучшается более 5 эпох меняем learning_rate на 0.0001)

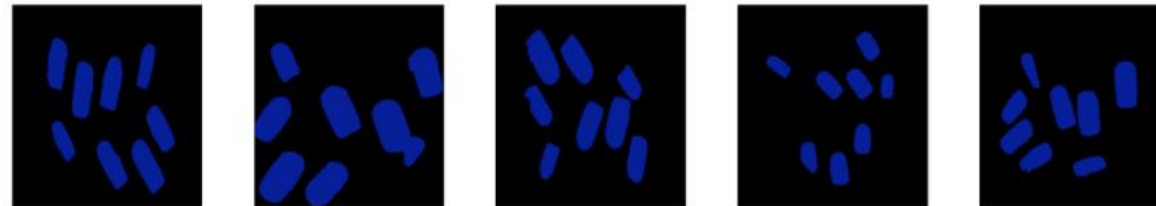
```
150/150 [=====] - 130s 868ms/step - loss: 0.0141 - dice_loss: 0.0085 - val_loss: 0.0356 - val_dice_loss: 0.0127  
Epoch 00067: early stopping  
Epoch with best val_dice_loss: 00057  
val_dice_loss: 0.012569504790008068  
Время обучения: 8761.48 с
```



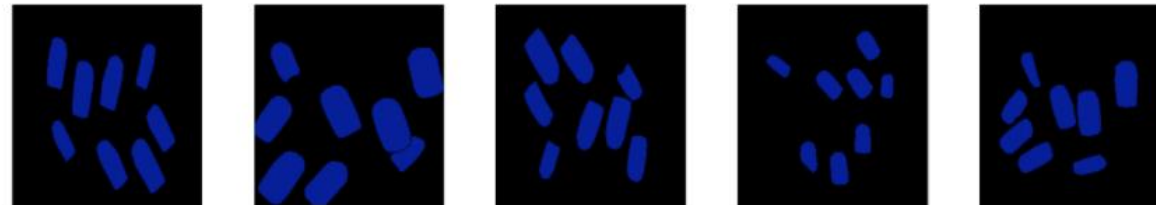
val_dice_loss: 0.0127

```
1 # Распознавание  
2 processImage(bss_relu_ep100_model_aug_fin, 2, 5) # функция отображения результатов обучения
```

Predict



Сегментация

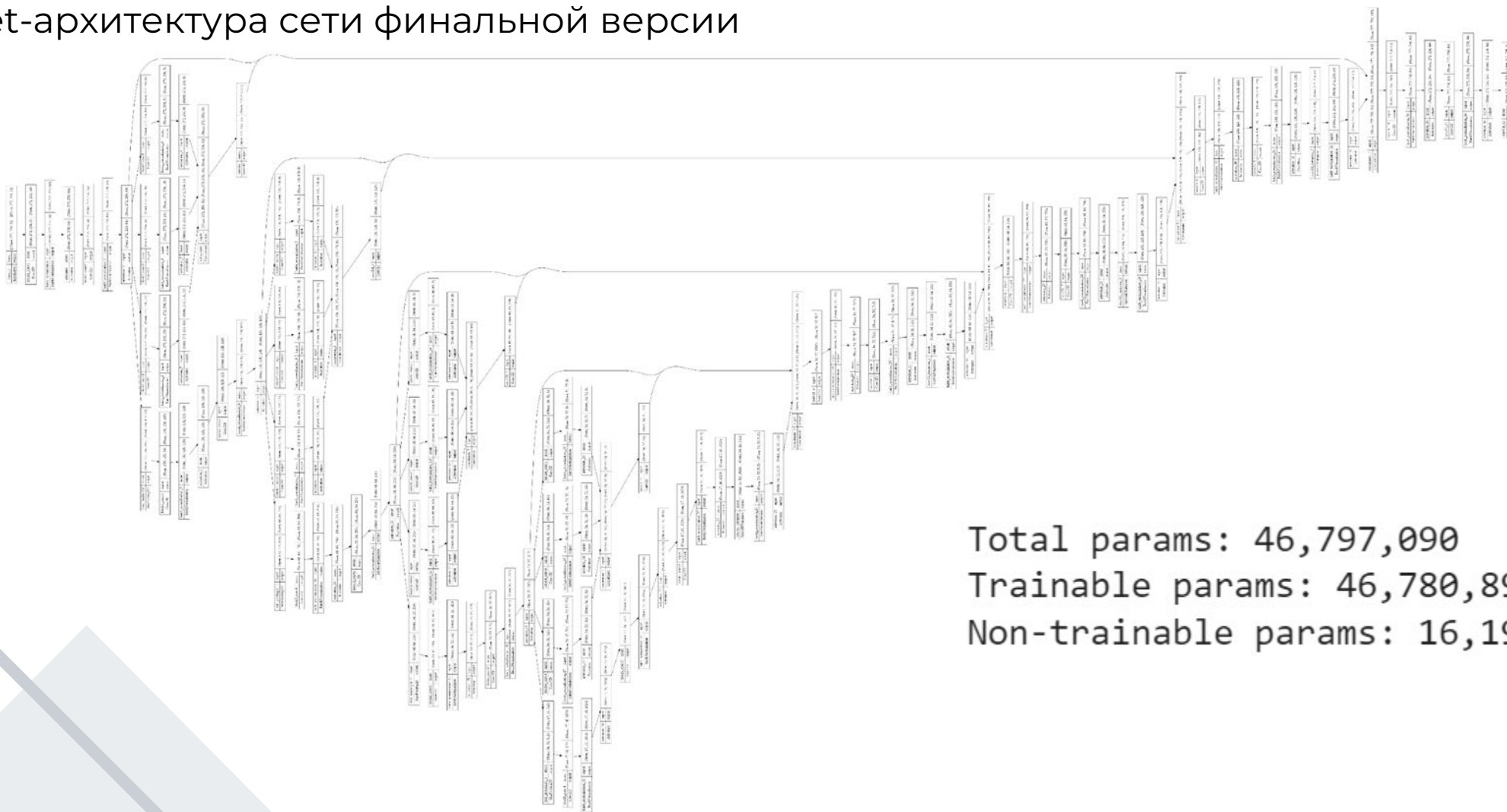


Исходное
изображение



Архитектура сети и её обучение

U-Net-архитектура сети финальной версии



Total params: 46,797,090

Trainable params: 46,780,898

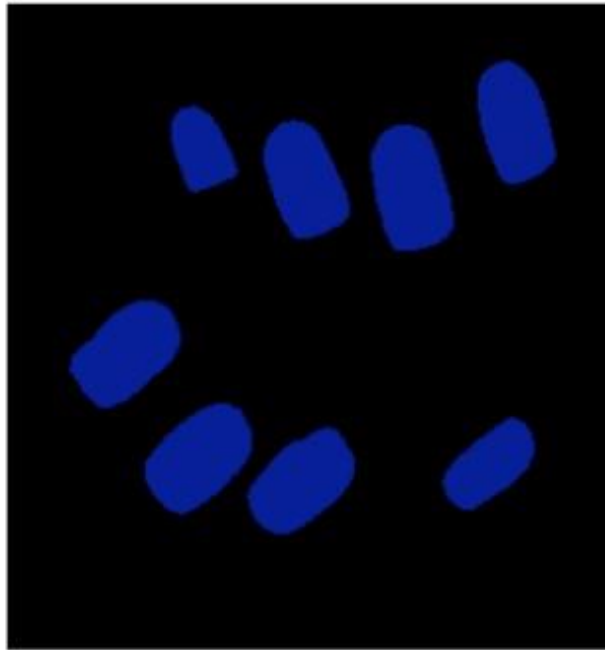
Non-trainable params: 16,192

Тест модели - наложение нового цвета маникюра

Исходное изображение



Predict



Изображение с новым цветом маникюра



В предикт подаём случайное фото из интернета, которого не было при обучении модели, распознаём области ногтей и перекрашиваем в новый цвет

Вывод

Результаты полученные при выполнении данной дипломной работы подтверждают тезисы о том, что применение U-Net-архитектур позволяют достичь высоких результатов в различных задачах сегментации изображений.

При этом, для достижения хороших результатов возможно использование небольшого количества данных, как в данной работе, где было использовано 400+ оригинальных изображений и с помощью аугментации, обучающие и проверочные данные были расширены до 1300+ примеров. Тем не менее были получены хорошие результаты.



Заключение

- Задачу дипломного проекта считаю выполненной
- Конечная цель - создание WEB-приложения «Дизайн маникюра с помощью ИИ» - будет достигнута при получении дополнительных знаний, не входящих в основную программу обучения