

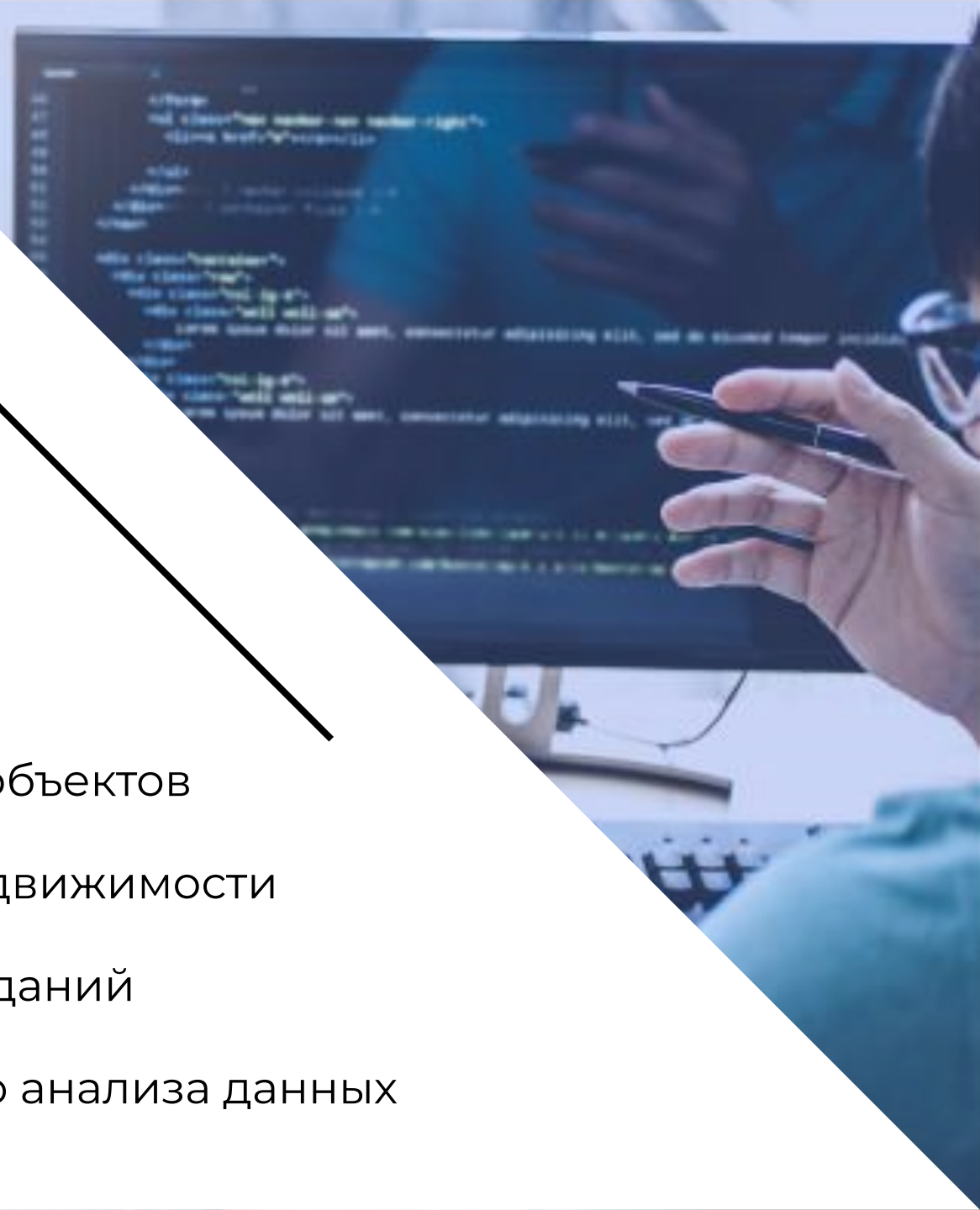
Сулим Андрей Сергеевич

Аналитика зданий

март 2023

Применение ИИ в сфере недвижимости

- Системы управления зданиями
- Прогнозирование спроса на недвижимость
- Прогнозирование стоимости недвижимости
- Оптимизация расположения строительных объектов
- Рекомендательные системы для покупки недвижимости
- Генеративные алгоритмы проектирования зданий
- Контроль качества строительства с помощью анализа данных



Задача:

Сегментация фасада здания и определение количества окон на нем

Подходы:

1. Применение архитектур U-Net или PSP-Net для сегментации фасада здания и окон на нем
2. Подсчет классов окон на сегментированном изображении

Обучающая база

Объем базы

360 изображений фасадов зданий
и карт сегментации

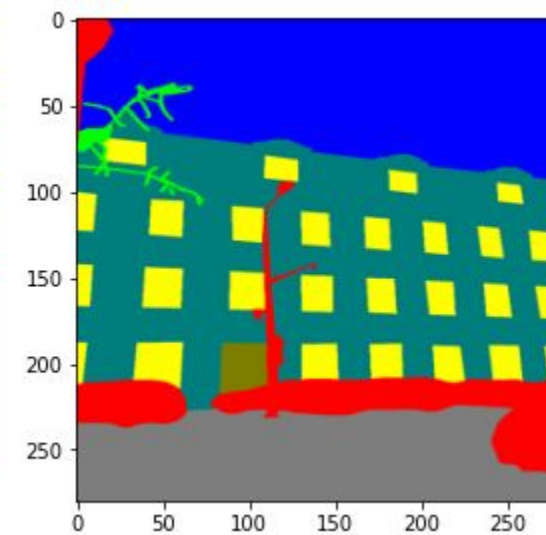
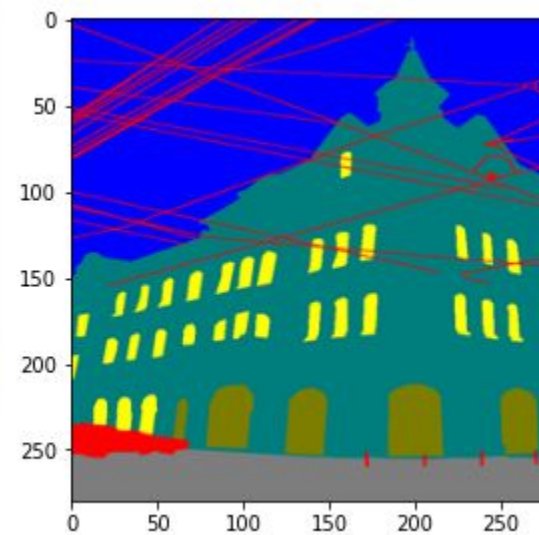
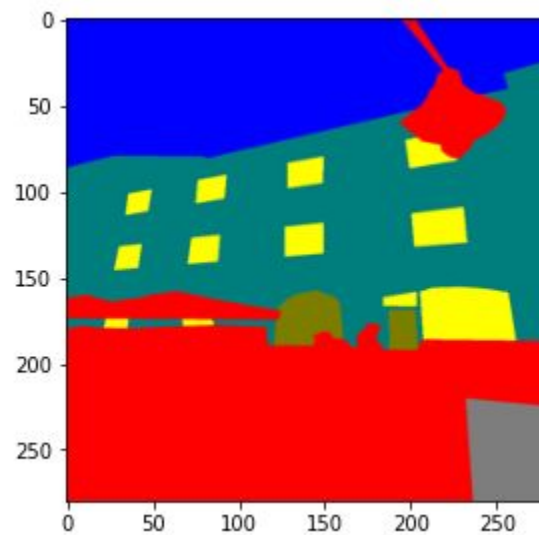
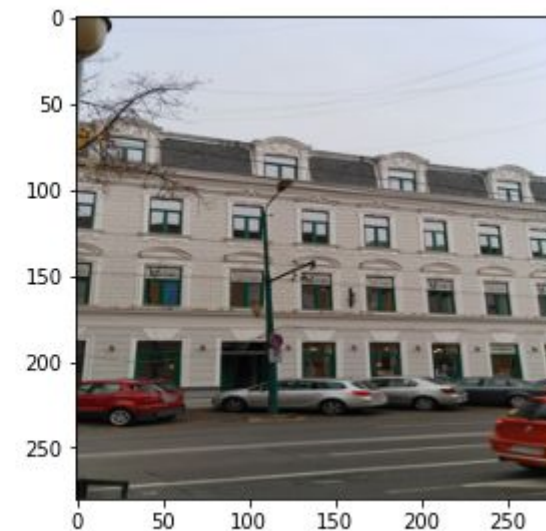
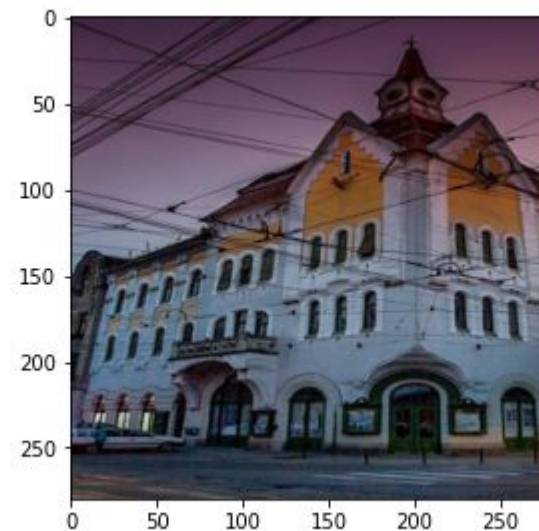
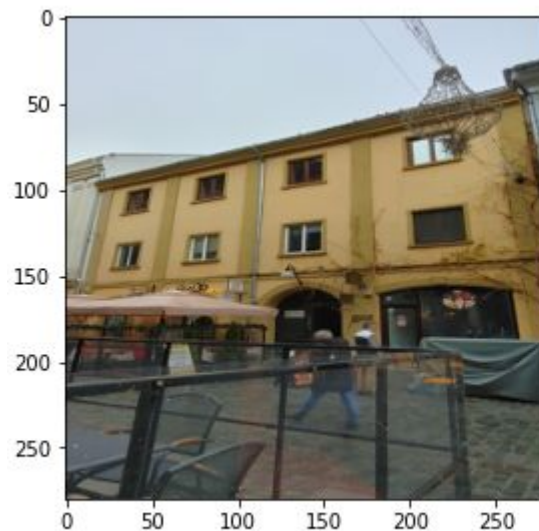
Источники сбора данных

http://www.ipb.uni-bonn.de/projects/etrimms_db

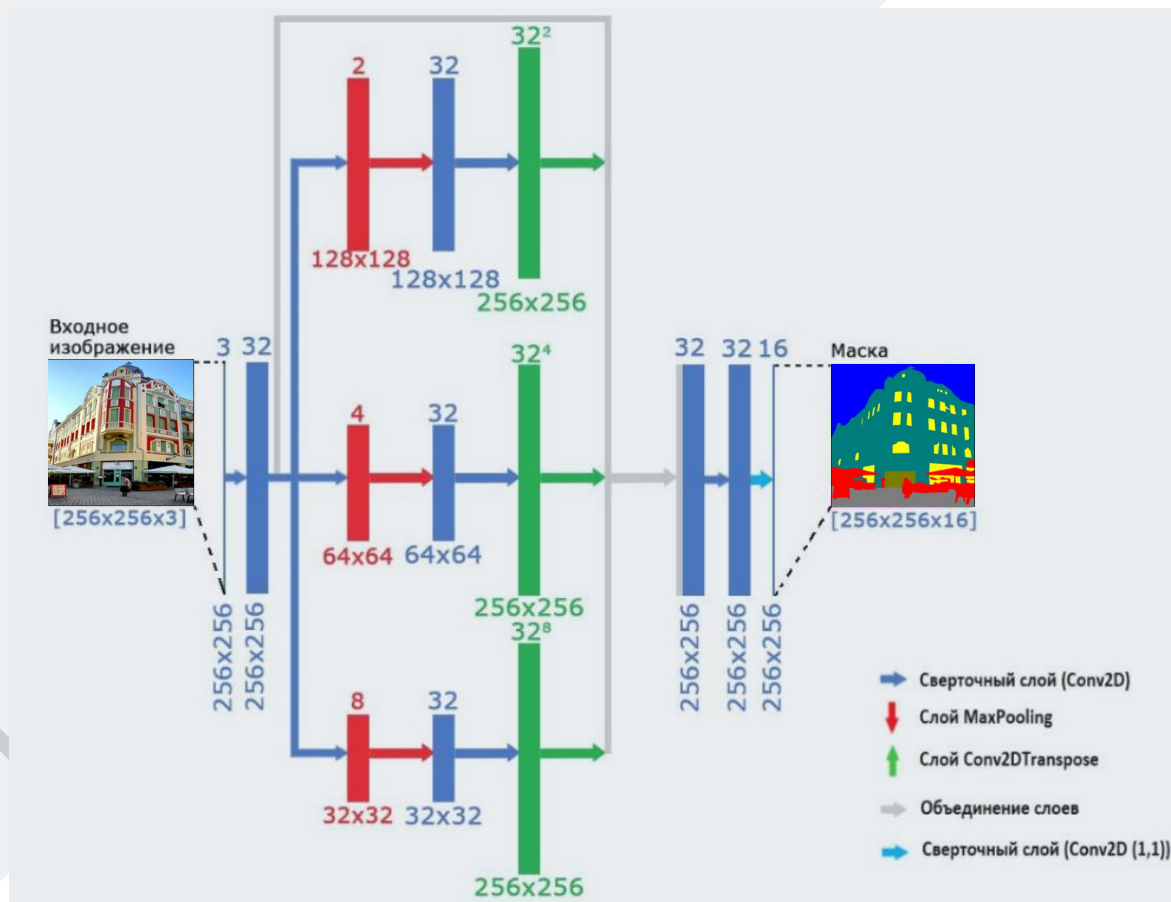
<https://github.com/CipiOrhei/TMBuD>



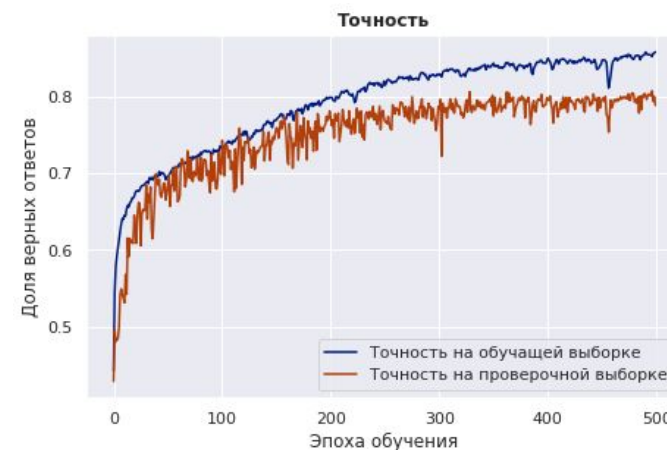
Пример изображений из базы



Архитектура PSP-Net



Результаты обучения модели

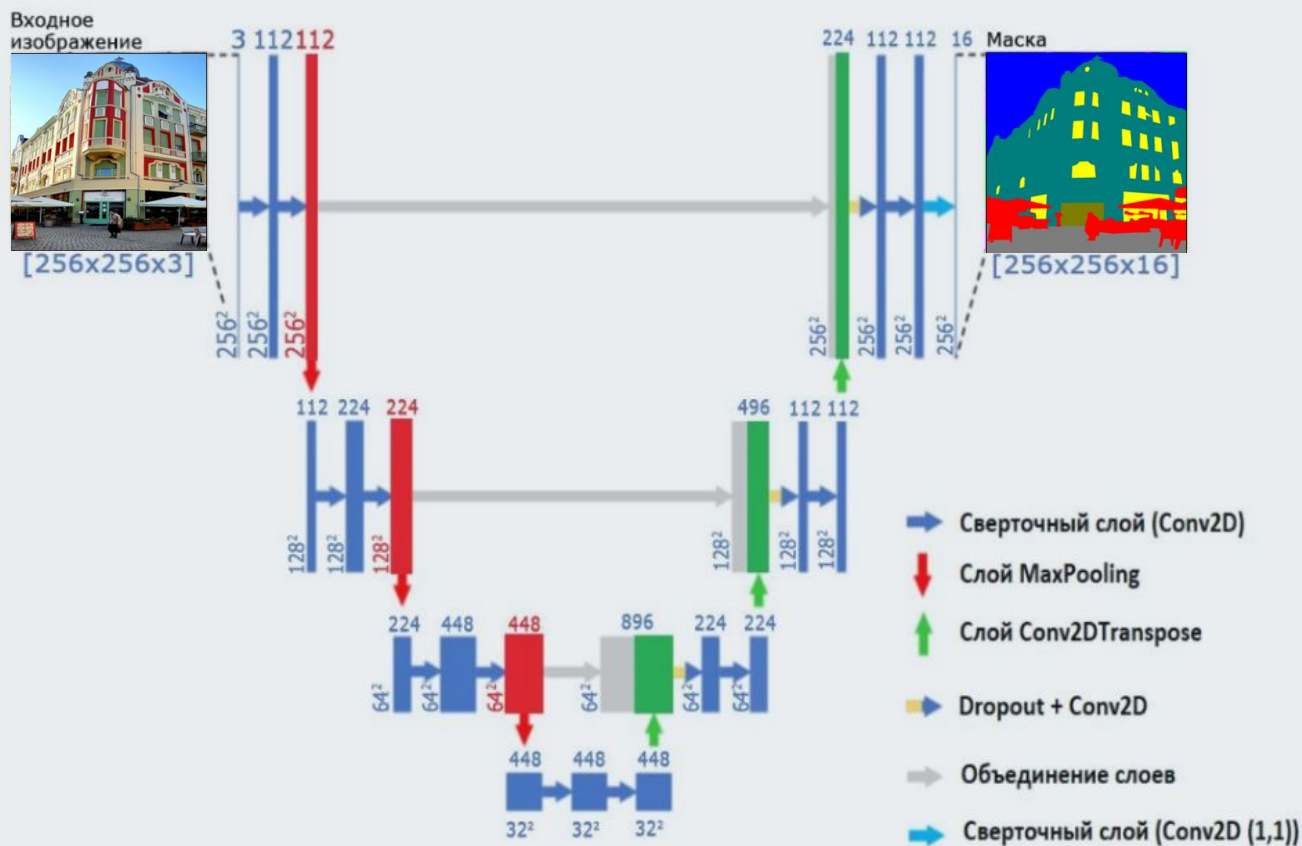


Максимально достигнутое значение точности:

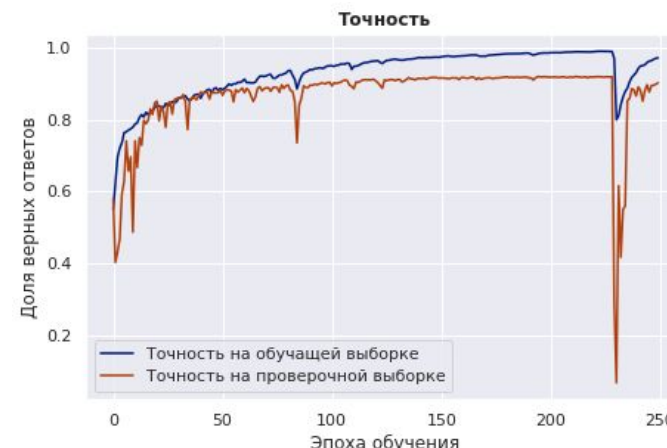
80,78 %



Архитектура U-Net



Результаты обучения модели



Максимально достигнутое значение точности:

91,95 % - после основного обучения

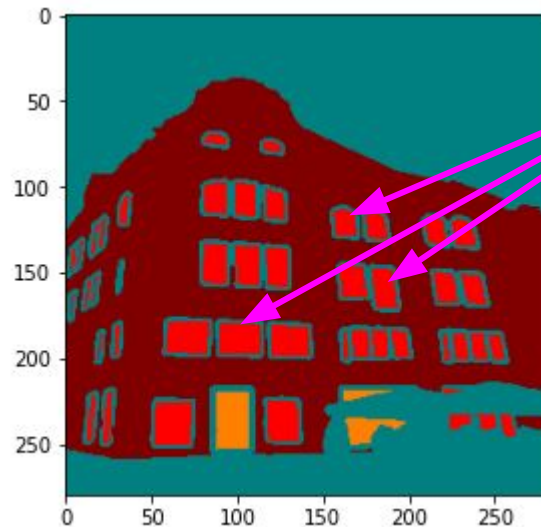


97,55 % - после дообучения с меньшим шагом learning_rate

Алгоритм нахождения количества окон в здании

Для нахождения количества окон, будем использовать предикт модели и считать количество областей нужного цвета

Для реализации этой задачи будем использовать библиотеку OpenCV для работы с изображениями.

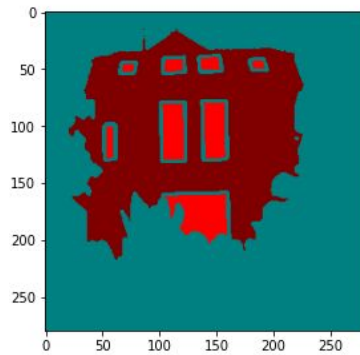


Ищем области с цветом [255, 0, 0]

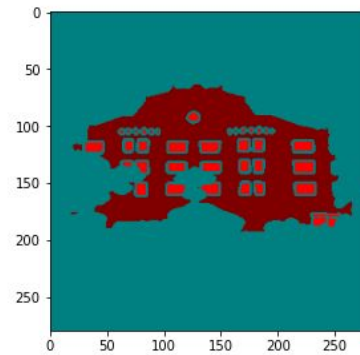
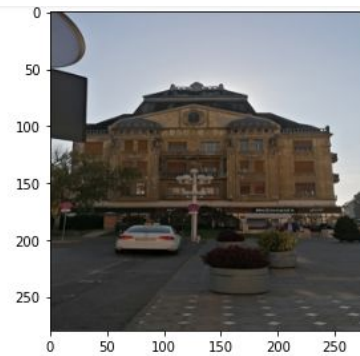
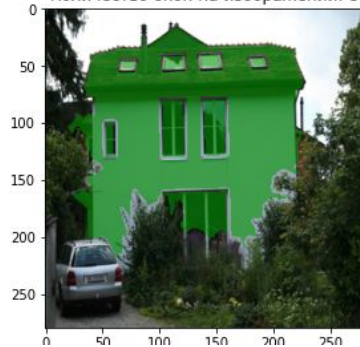
```
img = cv2.cvtColor(np.array(img.convert('RGBA')), cv2.COLOR_RGB2BGR)
color = np.array([255, 0, 0]) # Определение цвета, который нужно найти (в данном случае [255, 0, 0]) -
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB) # Изменение формата изображения, чтобы было удобнее работать
mask = cv2.inRange(img, color, color) # Создание маски для цвета [255, 0, 0]
contours, _ = cv2.findContours(mask, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE) # Поиск контуров областей
count = len(contours) # Подсчет количества областей
```


Проверка работы модели

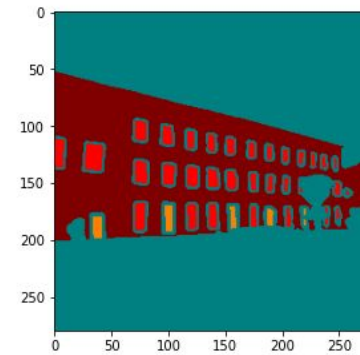
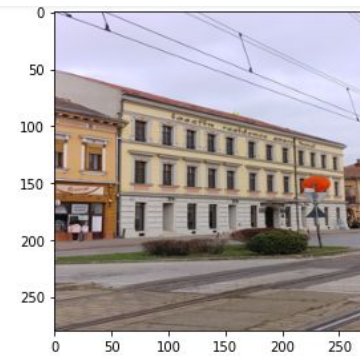
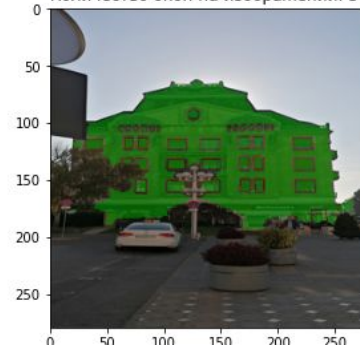
изображения проверочной выборки



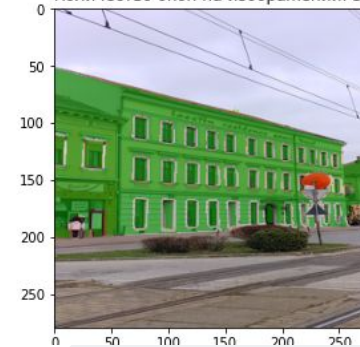
Количество окон на изображении: 8



Количество окон на изображении: 37

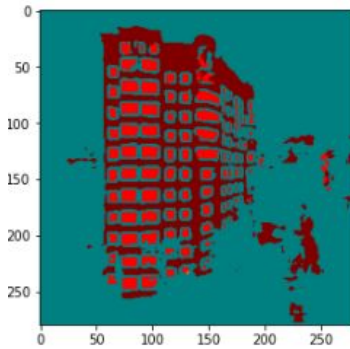
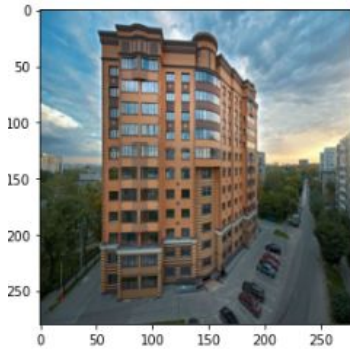


Количество окон на изображении: 34

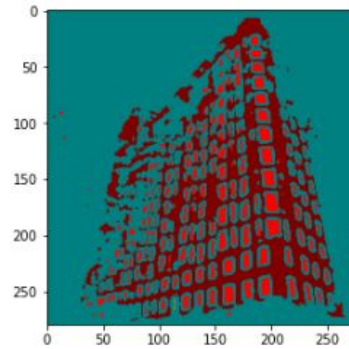
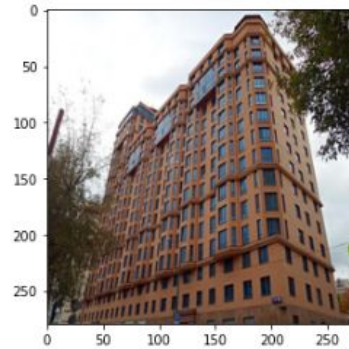
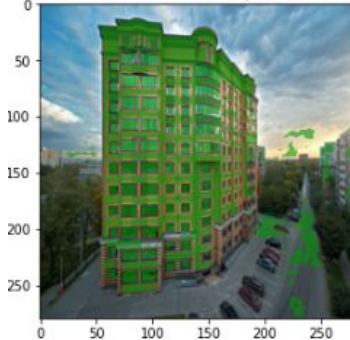


Проверка работы модели

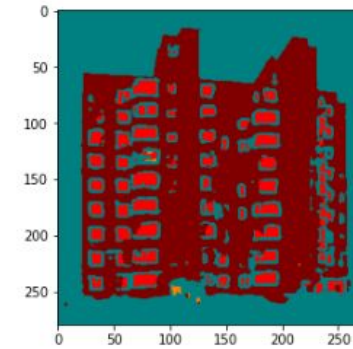
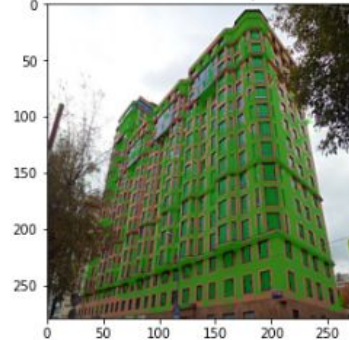
тестовые изображения многоквартирных домов из интернета



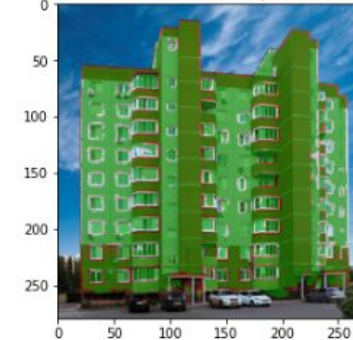
Количество окон на изображении: 124



Количество окон на изображении: 245



Количество окон на изображении: 116



Выводы

1. Для решений поставленной задачи наилучшую точность показала архитектура U-Net в сравнении с архитектурой PSP-Net
2. Целесообразно алгоритм подсчета количества окон реализовывать через функционал библиотеки OpenCV
3. Для корректного выделения фасада здания и подсчета количества окон многоквартирных домов обучающая база должна в большей степени состоять из фото именно многоквартирных домов и размечена соответствующим образом.

В работе использовалась база разных зданий, в меньшей степени относящихся к современным многоквартирным домам, и отличающихся от фото из интернета, на которых показана проверка.

The image features a prominent diagonal split between black and white. A thin grey line runs parallel to this split. The background is filled with a blurred, colorful image of computer code, likely from a terminal or IDE, with various colors like blue, yellow, and red highlighting different parts of the text. In the center, the word 'Спасибо' is written in a large, bold, black font.

Спасибо