Семантическая сегментация спутниковых снимков
на примере распознавания застроенных территорий

Работа выполнена: Семиной Анной

Введение

Автоматическое распознавание спутниковых снимков — это наиболее перспективный способ получения информации о расположении различных объектов на местности. Отказ от ручной сегментации снимков особенно актуален, когда речь заходит о обработке больших участков земной поверхности в сжатые сроки. Целью данной работы является построение модели на базе сверточных нейронных сетей для распознавания застроенных территорий на спутниковых снимках различных городов.

Благодаря автоматическому обнаружению зданий мы можем прогнозировать плотность населения или рассчитывать площадь жилых, коммерческих или нежилых земель в различных аналитических задачах.

Постановка задачи

Решается задача бинарной сегментации — на вход нейронной сети подаются цветные изображения (спутниковые снимки высокого разрешения), на которых необходимо выделить области пикселей, относящихся к одному классу — застроенные территории.

Описание данных

В работе использовался Inria Aerial Image Labeling Dataset (https://project.inria.fr/aerialimagelabeling/).

Изображения были получены во время нескольких летных кампаний над различными городскими районами США и Австрии. Маски для обучения моделей созданы путем растрирования шейп-файлов с информацией о застройке из открытых источников и состоят из двух классов: «здание» и «не здание». Набор данных содержит 180 спутниковых снимков в разрешении 5000x5000 пикселей и 180 масок.

В таблице 1 представлены регионы, включенные в обучающую выборку. На рисунке 1 приведен пример спутникового снимка и его маска.

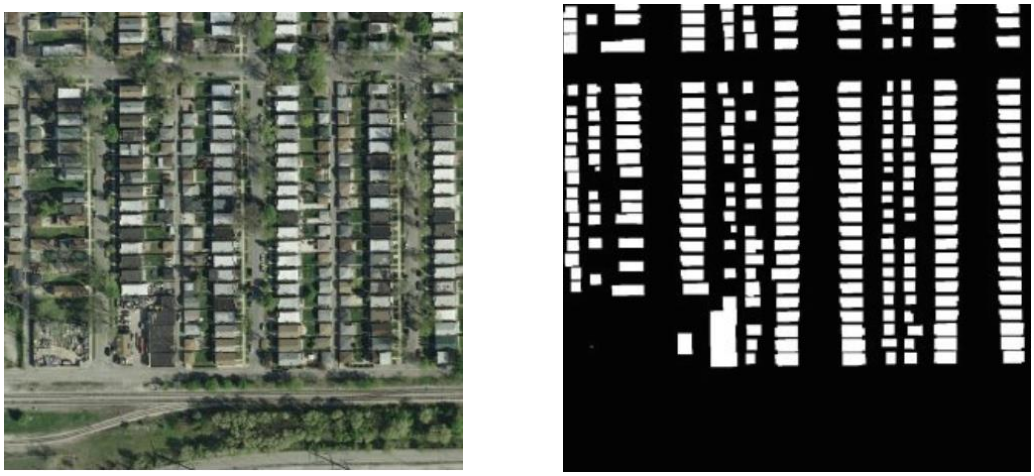


Рисунок 1 - Пример спутникового снимка (Чикаго) и его маски из набора данных для обучения

Город	Кол-во изображений	Общая площадь на изображениях
Остин, Техас	36	81 км^2
Чикаго, Иллинойс	36	81 км^2
Китсап, шт. Вашингтон	36	81 км^2
Вена, Австрия	36	81 км^2
Западный Тироль, Австрия	36	81 км^2
Всего	180	405 км^2

Таблица 1 - Набор данных для обучения модели

Обзор литературы

Был проведен анализ литературы и выявлены основные модели, используемые для сегментации спутниковых изображений. Наиболее часто в подобных задачах использовались модели FCN, U-Net, SegNet (Рисунок 3) [1].

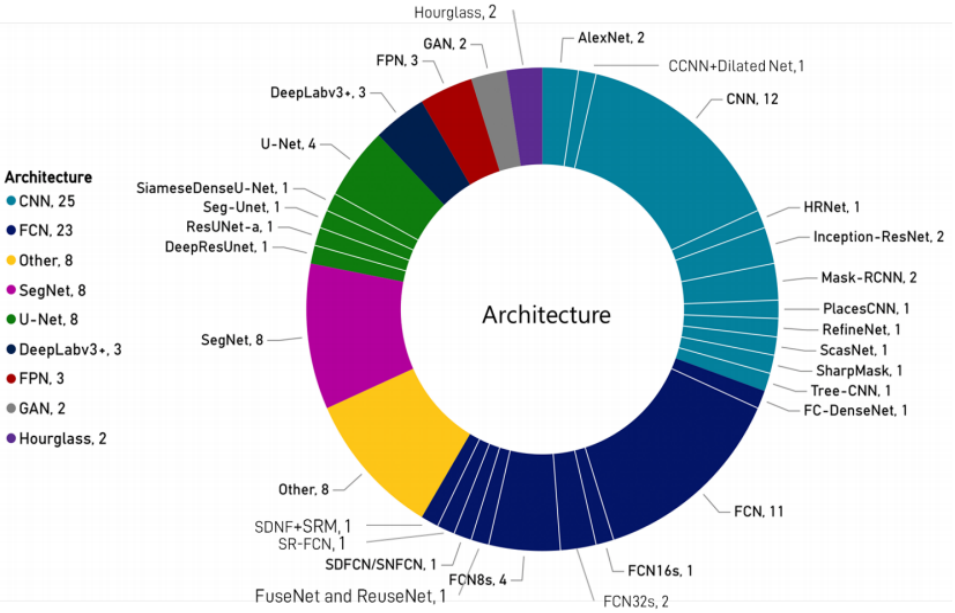


Рисунок 2 - Обзор используемых DL-архитектур [1]

Также был проведен анализ используемых базовых весов CN. Наиболее часто были использованы ResNet (ResNet-34, ResNet-50) и VGG (VGG, VGG-11, VGG-16, VGG-19) (Рисунок 4) [1].

Так же было обнаружено на примере U-net, что «пред-обученные» модели показывают лучший результат, чем «не пред-обученные» модели (Рисунок 5) [2, 3].

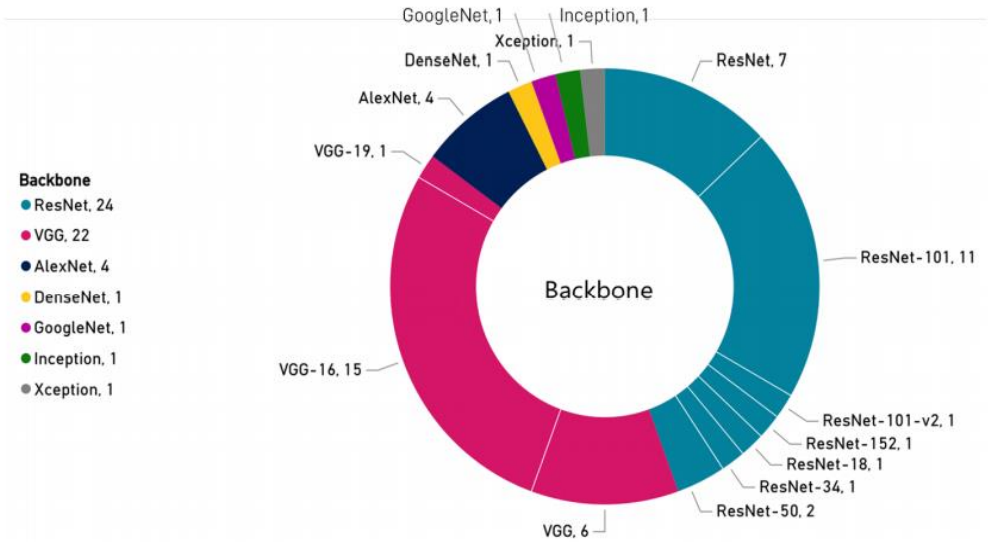
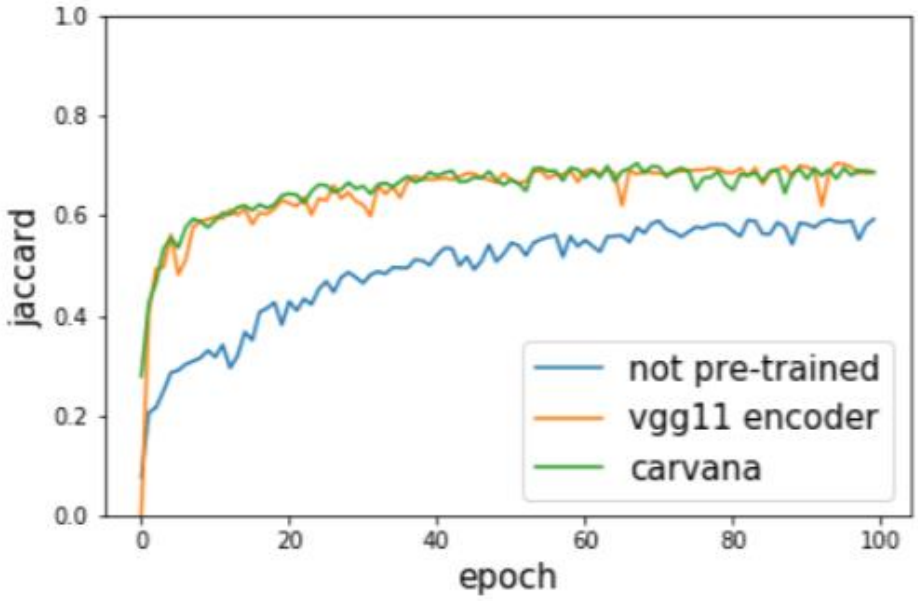


Рисунок 3 - Обзор используемых базовых CN [1]



Был проведен анализ подготовки исходных изображений к моделированию (Рисунок 7) [1].

Рисунок 4 - Анализ подготовки исходных изображений к моделированию [1]

Был проведен анализ метрик для оценки результатов сегментации. Наиболее часто использовались коэффициент Дайса (F1/Dice) и точность (Accuracy) (Рисунок 6) [1]. В исследуемых статьях у лучших моделей точность «Accuracy» достигала 91-93% [1, 5], коэффициент Жаккара стремился к 0.8 [1, 5], а коэффициент Дайса составлял 0.8-0.9 [1].

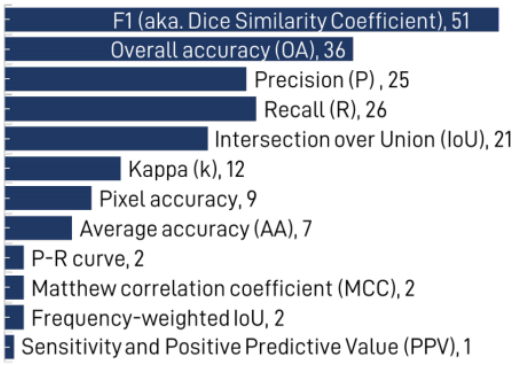


Рисунок 5 - Обзор используемых метрик [1]

Выбор метрики

В задачах сегментации одними из самых предпочтительных метрик являются коэффициент Жаккара [4],

$$J = \frac{TP}{TP + FP + FN} = \frac{A \cap B}{A \cup B} = \frac{A \cap B}{A + B - A \cap B}$$

который измеряет сходство между правильной (обучающей) разметкой и предсказанной моделью, и определяется как размер пересечения, деленный на размер объединения наборов этих выборок (Рисунок 8).

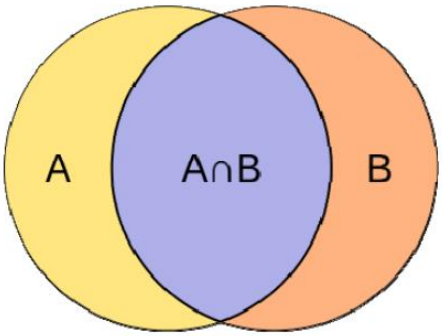


Рисунок 6 - Пояснение к описанию коэффициента Жаккара

Похожий смысл имеет и коэффициент Дайса (Sørensen–Dice coefficient). Он отличается от индекса Жаккара, который учитывает True Positive результаты только один раз как в числителе, так и в знаменателе.

$$DSC = \frac{2|X \cap Y|}{|X| + |Y|} = \frac{2TP}{2TP + FP + FN}$$

Для оценки сегментации в нашей работе использовался коэффициент Дайса.

Предобработка данных

Было использовано несколько способов предобработки данных.

1) Уменьшение разрешения изображений и масок с 5000*5000 до 512*512. Сделано это было в надежде упростить работу в среде Colab (из-за ограниченного времени обучения и ограниченного места на диске). Данный способ очень сильно снизил качество обучения.

2) Нарезание изображений 224*224 из исходных данных. Для этой целей была написана программа cropping.py (см. репозиторий), которая из каждого изображения и каждой маски вырезает квадраты, главная диагональ которых имеет следующее расположение (Рисунок 2):

    - от (1025; 1025) px до (1249; 1249) px
    - от (4225; 4225) px до (4449; 4449) px
    - от (1025; 4225) px до (1249; 4449) px
    - от (2225; 2225) px до (2449; 2449) px

Рисунок 7 - Пример нарезания изображений из исходного и результат нарезания

Разделение на выборки для обучения и валидации происходили следующим образом: так как каждый город содержал по 36 изображений и каждое из них было пронумеровано, то на валидационную выборку отводились изображения с порядковым номером более 31 (то есть 25 изображений, по 5 для каждого города). В этом случае обучающая выборка содержала 155 изображений. Для нарезанных картинок нумерация сохранялась и валидационный набор попадали вырезанные картинки с тех же изображений. Получалось 620 изображений в обучающей выборке и 100 в валидационной. Для модели TernausNet обучающая и валидационная выборки генерировались рандомным образом так, что для валидации отбиралось 25 картинок, а остальные картинки нарезались (по 5 образцов заданного размера с каждой).

Выбор моделей

В проекте были выбраны следующие модели:

1. UNET_Resnet34 (из FastAI)
2. UNET_VGG16 (из FastAI)
3. UNET_VGG (из Keras)
4. Модифицированная TernausNet (базируется на UNET_VGG16).

Результаты моделирования

1. VGG_UNET (из keras_segmentation.models)

Эта модель была так называемой «пробой пера». Результирующая точность составила 71,1%. Процесс обучения показан на рисунке 9. Модель отмечает области застройки, но на предсказанной сегментации можно заметить довольно много «шума» (Рисунок 10).

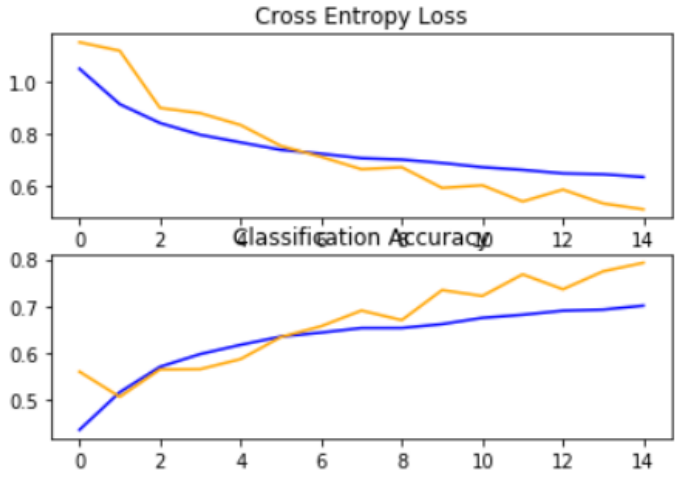


Рисунок 8 - Процесс обучения VGG_UNET

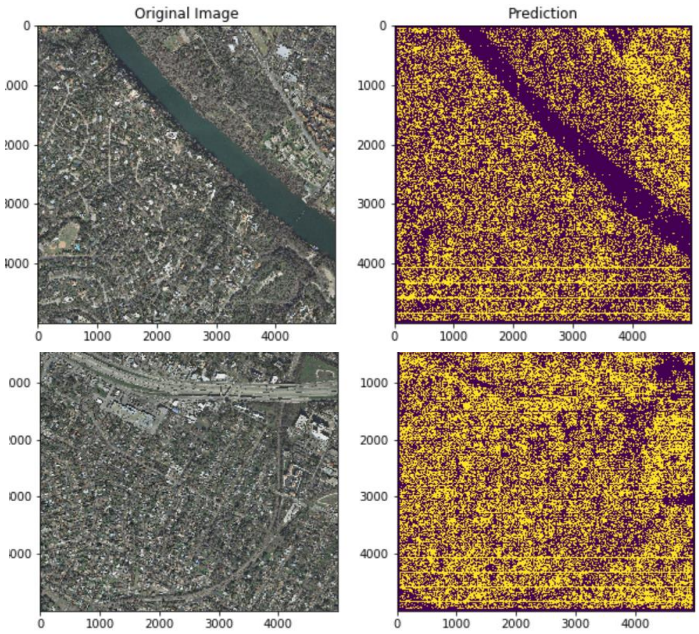


Рисунок 9 - Пример сегментации VGG_UNET

2. UNET_Resnet34

Моделирование проводилось в системе Colab с использованием библиотеки FastAI. Изначально, из-за ограниченного места на диске и ограниченного времени на обучение, было принято «неудачное» решение сжать тренировочную выборку и маски с 5000x5000 до 512x512. Процесс обучения представлен на рисунке 11. Видно, что «точность» уже после нескольких эпох возросла до 90%, а «Dice» в течение всего обучения колебалась около 0.2. Глядя на предсказание сегментации можно сделать вывод, как мало большая точность значит в задачах сегментации. Модель смогла распознать только большие здания и полностью проигнорировала мелкую сельскую застройку (Рисунок 12).

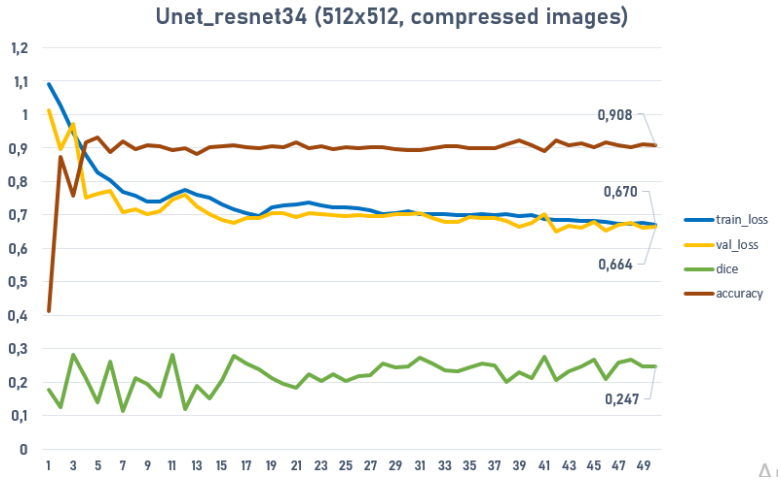


Рисунок 10 - Процесс обучения Unet_resnet34 (compressed)

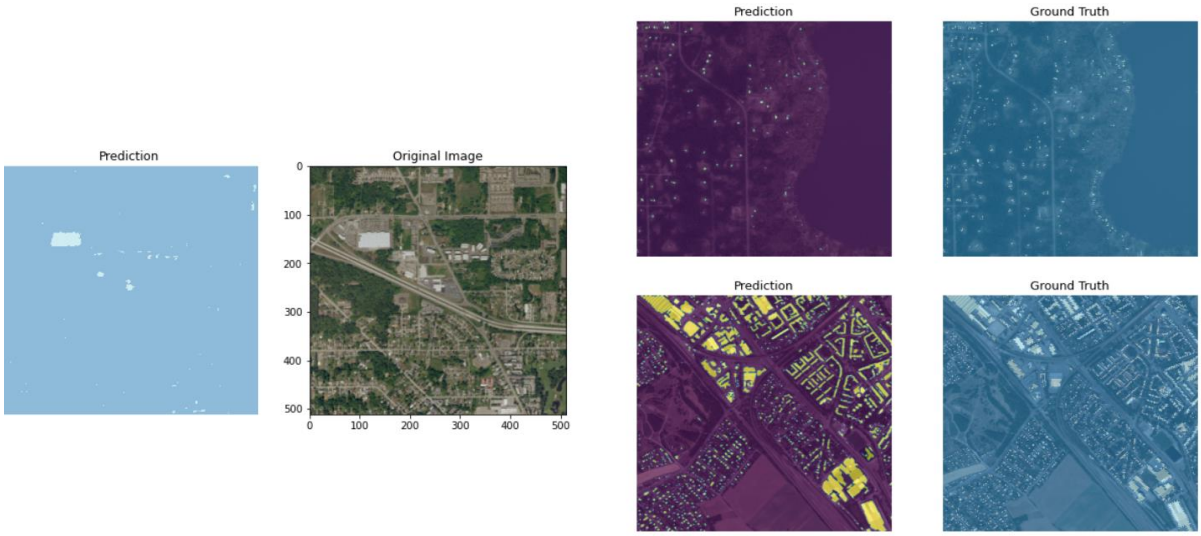


Рисунок 11 – Пример сегментации, полученной Unet_resnet34 (compressed) на тестовой выборке (слева) и обучающей выборке (справа)

Результирующие метрики **Unet_resnet34 (compressed)** приведены в таблице 2.

Далее, исправились, и запустили эту же модель на вырезанных изображениях (см. главу «Предобработка данных»). В этом случае «Dice» вырос до 0,35. Процесс обучения представлен на рисунке 13. Стала немного определяться мелкая сельская застройка, но в

предсказаниях на обучающей выборке заметен «некоторый шум» на незастроенных участках (Рисунок 14).

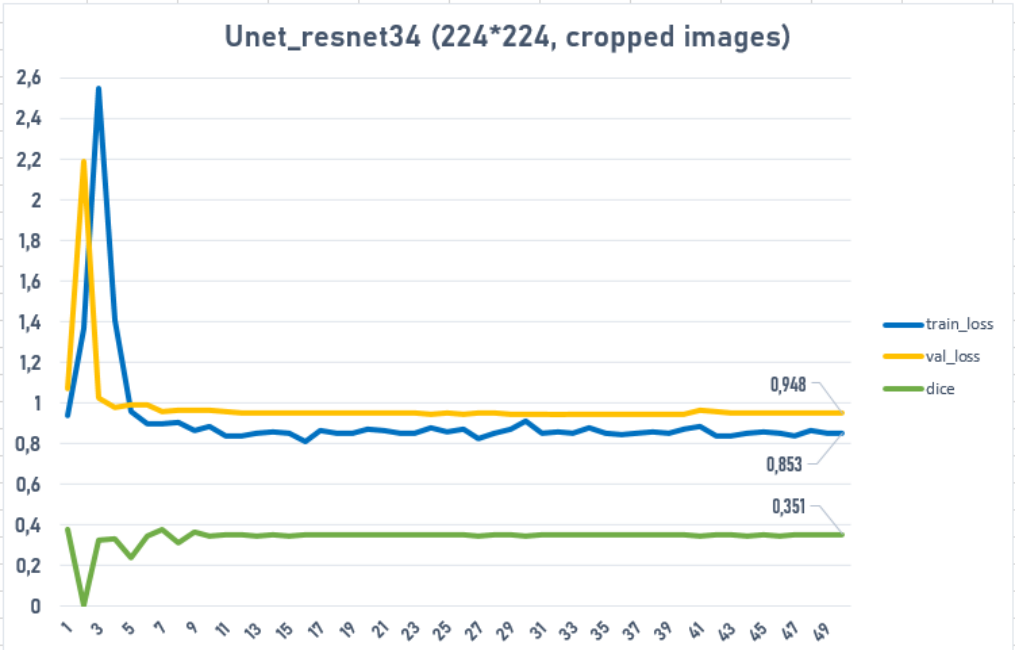


Рисунок 12 - Процесс обучения Unet_resnet34 (cropped)

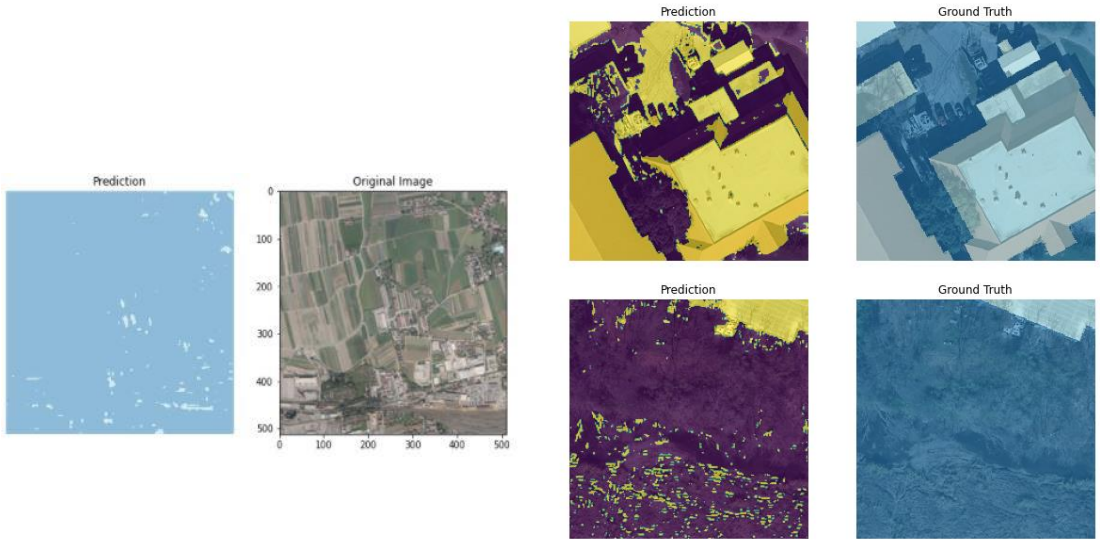


Рисунок 13 - Пример сегментации, полученной Unet_resnet34 (cropped) на тестовой выборке (слева) и обучающей выборке (справа)

Результирующие метрики **Unet_resnet34(cropped)** приведены в таблице 2.

3. UNET_VGG16

Моделирование также проводилось в системе Colab с использованием библиотеки FastAI. Также сначала было использовано «неудачное» решение сжать тренировочную выборку и маски с 5000x5000 до 512x512. Процесс обучения представлен на рисунке 15. Итоговый

«Dice» оказался ниже, чем у resnet34 в аналогичном случае, и составил чуть больше, чем 0,1. Модель также смогла распознать только большие здания и проигнорировала мелкую сельскую застройку (Рисунок 16).

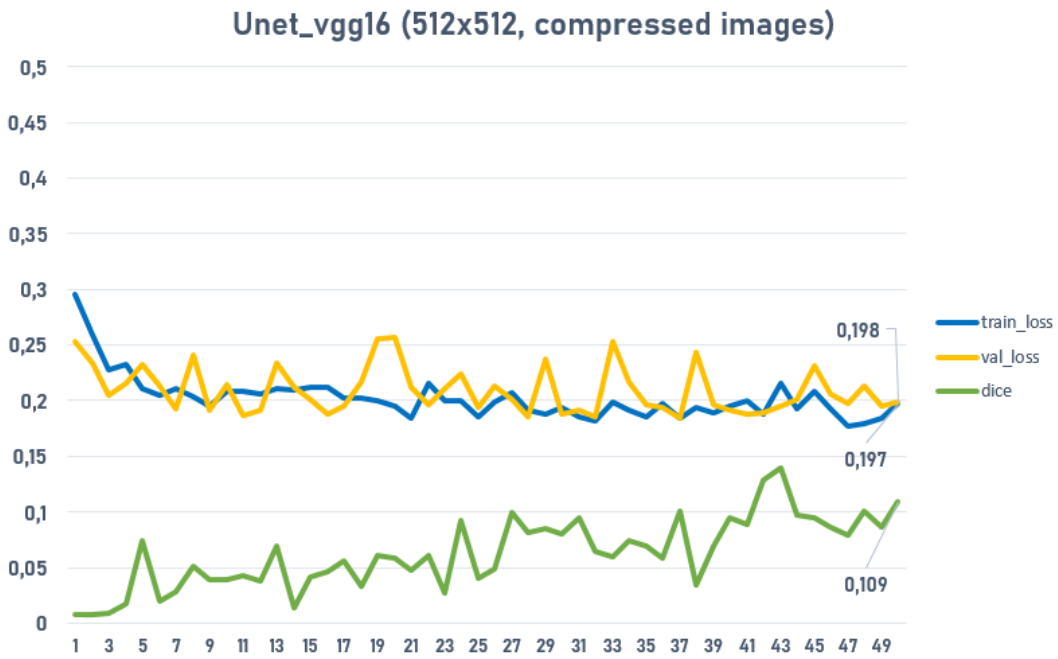


Рисунок 14- Процесс обучения Unet_VGG16 (compressed)

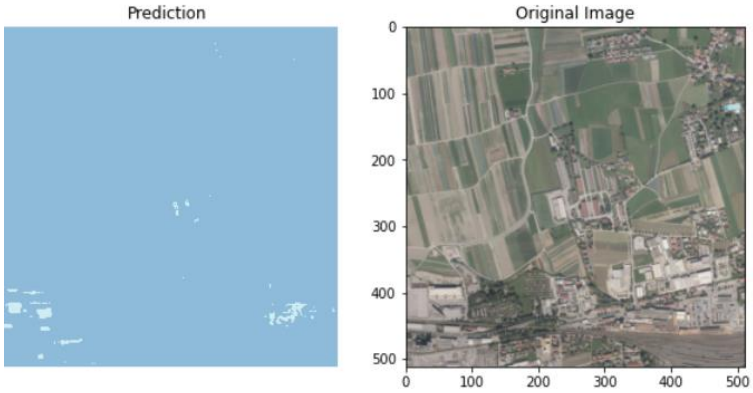


Рисунок 15 - Пример сегментации, полученной Unet_VGG16 (compressed)

Результирующие метрики **Unet_VGG16 (compressed**) приведены в таблице 2.

Запустили эту же модель на нарезанных изображениях (см. главу «Предобработка данных»). Итоговый «Dice» получился 0,682, в два раза больше, чем у аналогичной resnet34. Процесс обучения представлен на рисунке 17. Качественный анализ сегментации, производимой этой моделью на тестовом изображении (тестовое изображение имеет исходный размер 5000*5000) тоже вполне удачен, была отмечена как крупная, так и не очень крупная застройка (Рисунок 18).

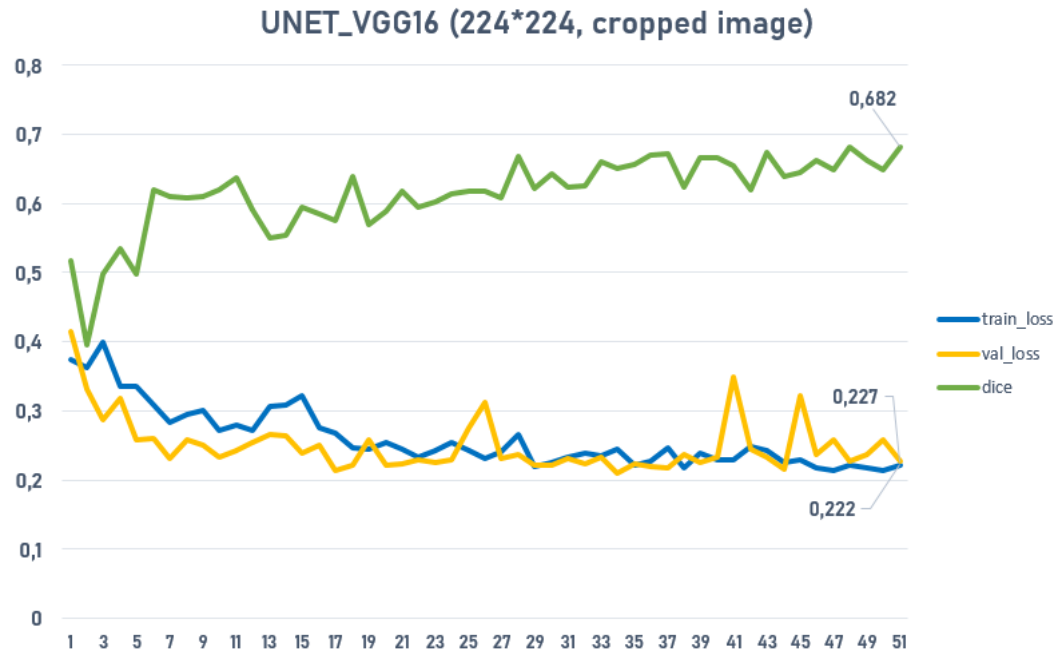


Рисунок 16 - Процесс обучения Unet_VGG16 (cropped)

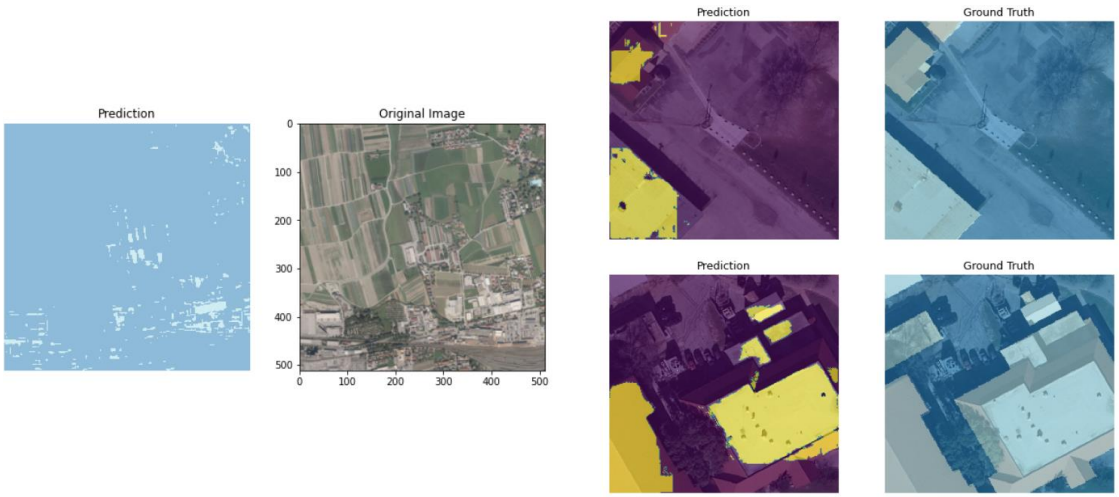


Рисунок 17 - Пример сегментации, полученной Unet_VGG16 (cropped) на тестовой выборке (слева) и обучающей выборке (справа)

Результирующие метрики **Unet_VGG16 (cropped**) приведены в таблице 2.

4 TernausNet16 [3]

Для этой модели был использован немного другой способ предобработки данных. Была использована сериализация при помощи библиотеки «Pickle». С ее помощью при «нарезании» данных так же сохранялось встроенное в исходное TIFF изображение геопозиционирование. Итоговый «Dice» получился 0,598, чуть ниже, чем у аналогичной VGG16. Процесс обучения представлен на рисунке 19. При этом качество сегментации у этой модели очень хорошее, нет шума, довольно точно ловится как мелкая сельская застройка, так и крупные дома. (Рисунок 20)

Рисунок 18 - Процесс обучения TernausNet16 (cropped)

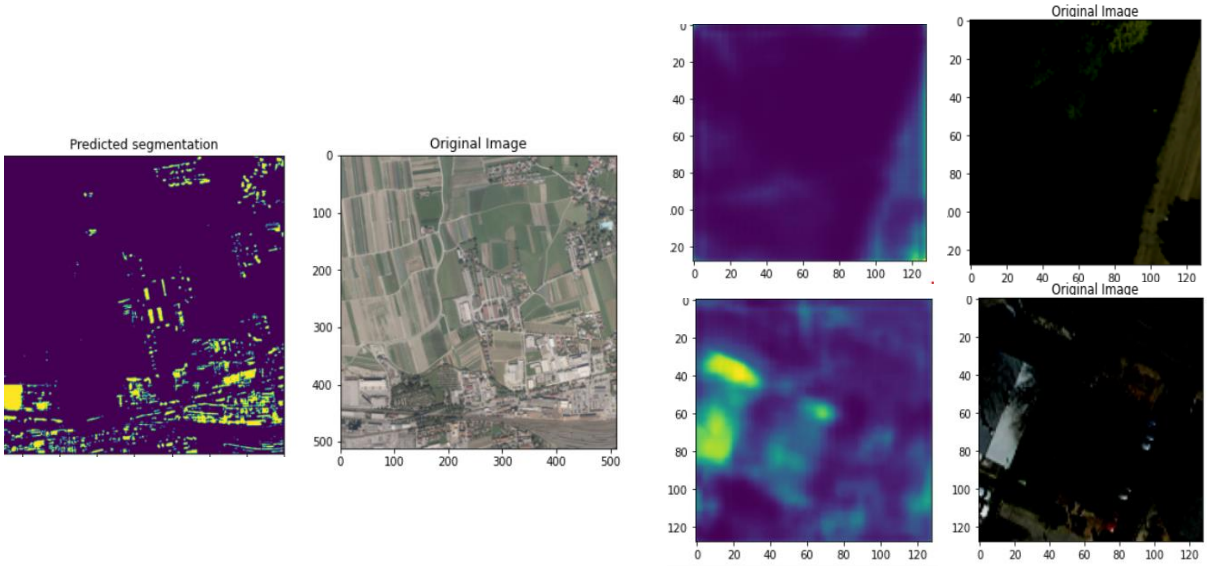


Рисунок 19 - Пример сегментации, полученной TernausNet16 (cropped) на тестовой выборке (слева) и обучающей выборке (справа)

Результирующие метрики **TernausNet16 (cropped**) приведены в таблице 2.

	U_ResNet34 (compressed)	U_ResNet34 (cropped)	U_VGG16 (compressed)	U_VGG16 (cropped)	TernausNet16 (cropped)
Train_loss	***0,665***	***0,853***	***0,198***	***0,222***	***0,301***
Val_loss	***0,658***	***0,948***	***0,197***	***0,227***	***0,298***
Accuracy	***0,912***	**-**	**-**	**-**	**-**
Dice	***0,243***	***0,351***	***0,109***	***0,682***	***0,598***

Таблица 2 – Результирующие метрики моделей, представленных в работе

Выводы

Было разработано несколько моделей, решающих задачу семантической сегментации спутниковых снимков. У лучших моделей коэффициент Дайса приближался к 0.7, что сравнимо с метриками из обзора литературы.

Качественная оценка сегментации, производимой моделями из данной работы, позволяет считать их применимыми для разметки спутниковых снимков. Конечно,есть некоторые погрешности: не всегда распознается мелкая сельская застройка, линия побережья размечается некоторыми моделями, как застроенная территория. Последнюю проблему можно решить более удачным подбором обучающей выборки.

В качестве продолжения работы над этим проектом можно расширить сегментацию на другие объекты на спутниковых снимках: дороги, леса, поля, реки.

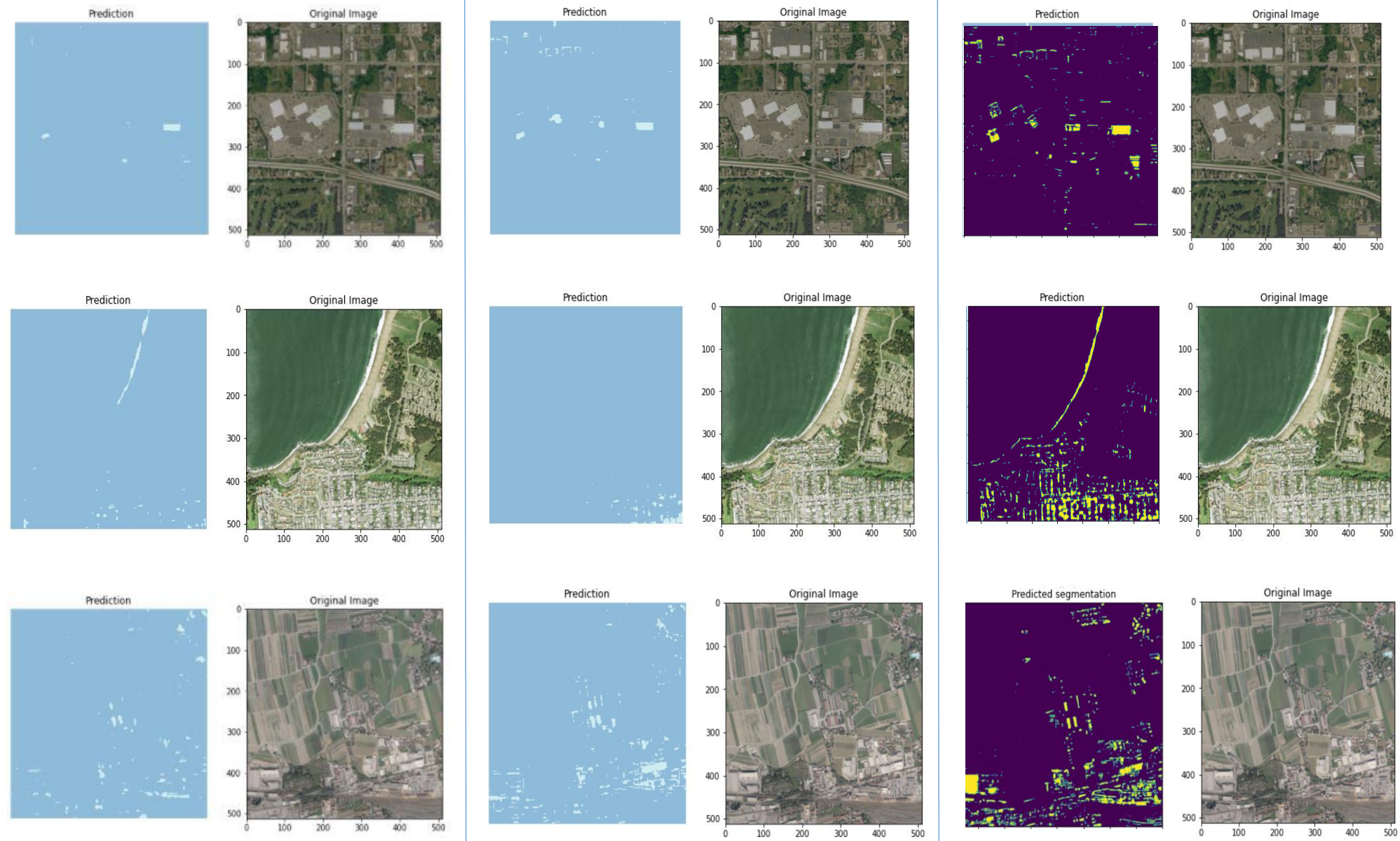


Рисунок 20 - Сравнение результатов сегментации лучших моделей

Список используемой литературы

1. Bipul Neupane, Teerayut Horanont and Jagannath Aryal, **Deep Learning-Based Semantic Segmentation of Urban Features in Satellite Images: A Review and Meta-Analysis**, *Remote Sens. 2021, 13, 808* (https://doi.org/10.3390/rs13040808)
2. Bohao Huang, Kangkang, Nicolas Audebert, Andrew Khalel, Yuliya Tarabalka, **Large-Scale Semantic Classification: Outcome Of The First Year Of Inria Aerial Image Labeling Benchmark**, *ResearchGate Conference Paper, July 2018* (DOI: 10.1109/IGARSS.2018.8518525)
3. Vladimir Iglovikov**, TernausNet: U-Net with VGG11 Encoder Pre-Trained on ImageNet for Image Segmentation**, *Cs.Cv, January 2018* (arXiv:1801.05746v1)
4. Р.А. Соловьев, Д.В. Тельпухов, А.Г. Кустов, **Автоматическая сегментация спутниковых снимков на базе модифицированной свёрточной нейронной сети UNET,** *Инженерный вестник Дона, №4 (2017)* (ivdon.ru/ru/magazine/archive/n4y2017/4433)
5. Wuttichai Boonpook , Yumin Tan, Yinghua Ye , Peerapong Torteeka **A Deep Learning Approach on Building Detection from Unmanned Aerial Vehicle-Based Images in Riverbank Monitoring** *Sensors 2018, 18, 3921* (doi:10.3390/s18113921)