



МИНОБРНАУКИ РОССИИ  
федеральное государственное бюджетное образовательное  
учреждение  
высшего образования  
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ  
ЭКОНОМИЧЕСКИЙ  
УНИВЕРСИТЕТ»  
(СПбГЭУ)

Факультет информатики и прикладной математики  
Кафедра прикладной математики и экономико-математических методов

**ОТЧЕТ**

**по производственной практике: научно-исследовательская работа**

Наименование организации прохождения практики: ООО «ПЛАЗ»

Направление (специальность) 01.03.02 Прикладная математика и информатика  
(код, наименование)

Направленность (специализация) Прикладная математика и информатика в  
экономике и управлении

Обучающийся Сафонов Дмитрий Андреевич  
(Ф.И.О. полностью)

Группа ПМ-1701  
(номер группы)

Подпись 

Руководитель  
практики от СПбГЭУ: Лебедева Людмила Николаевна, доцент,  
кандидат физико-математических наук  
(Ф.И.О., ученая степень, ученое звание)

\_\_\_\_\_  
(подпись руководителя)

Оценка по итогам защиты  
отчета \_\_\_\_\_

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
(подписи членов комиссии)

Санкт-Петербург  
2020 г.



### Личный листок

**ФИО обучающегося: Сафонов Дмитрий Андреевич**

**Факультет информатики и прикладной математики**

Курс обучения: 3, группа: ПМ-1701

Направление: 01.03.02 Прикладная математика и информатика

Направленность (профиль): Прикладная математика и информатика в экономике и управлении

Сроки практики: с 09.07.2020 г. по 22.07.2020 г.

Декан факультета: Барабанова Марина Ивановна

(Ф.И.О., подпись)

М.П.

---

Отметка о прохождении практики обучающегося Сафонова Дмитрия Андреевича в ООО «ПЛАЗ» с 09.07.2020 г. по 22.07.2020 г.

Обучающийся прошёл инструктаж по ознакомлению с требованиями охраны труда, техники безопасности, пожарной безопасности, а также с правилами внутреннего распорядка. Вводный инструктаж и инструктаж на рабочем месте пройдены с оформлением установленной документации.

Руководитель практики от организации/профильной организации назначен приказом №642 дата 09.07.2020ыы

Юрецкий Алексей Владимирович, заместитель генерального директора  
(Ф.И.О. должность)

М.П.

(подпись)



**МИНОБРНАУКИ РОССИИ**  
**федеральное государственное бюджетное образовательное**  
**учреждение**  
**высшего образования**  
**«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭКОНОМИЧЕСКИЙ**  
**УНИВЕРСИТЕТ»**  
**(СПбГЭУ)**

Факультет информатики и прикладной математики

Кафедра прикладной математики и экономико-математических методов

СОГЛАСОВАНО:

Руководитель практики от профильной  
организации

УТВЕРЖДАЮ:

Заведующий кафедрой прикладной математики  
и экономико-математических методов

Юрецкий Алексей Владимирович, заместитель  
генерального директора

Фридман Григорий Морицович  
(Ф.И.О.)

—  
(подпись)

«25» июня 2020 г.

М.П.

—  
(подпись)

«25» июня 2020 г.

## **Индивидуальное задание**

**для прохождения производственной практики**  
**(научно-исследовательская работа)**

**Обучающегося:** 3 курса Сафонова Дмитрия Андреевича

**Направление:** 01.03.02 Прикладная математика и информатика

**Направленность (профиль):** Прикладная математика и информатика в экономике и  
управлении

**Наименование организации прохождения практики:** ООО «ПЛАЗ»

**Сроки прохождения практики:** с 09.07.2020 г. по 22.07.2020 г.

**Форма предоставления на кафедру выполненного задания:** отчёт

**Дата выдачи задания:** 25.06.2020

**Руководитель практики от СПбГЭУ:** Лебедева Людмила Николаевна, доцент, кандидат  
физико-математических наук

### Совместный рабочий график (план проведения практики)

№ п/п	Перечень заданий, подлежащих разработке на практике	Календарные сроки (даты выполнения)
1.	Ознакомление с локальными нормативными актами предприятия, регламентирующими деятельность сотрудников, правилами внутреннего распорядка и безопасности труда, прохождение необходимых инструктажей.	09.07.20
2.	Ознакомление со структурой организации, её подразделениями и видами деятельности.	09.07.20
3.	Изучение программного обеспечения, используемого в организации, и применяемых методов автоматизации производства.	09.07.20
4.	Изучение и разметка данных аэросъемки беспилотными летательными аппаратами для дальнейшего решения задачи.	10.07.20
5.	Поиск дополнительных наборов данных для решения задачи распознавания людей на изображениях с аэросъемки.	11-14.07.20
6.	Изучение современных методов компьютерного зрения для решения задач сегментации и распознавания образов.	15-17.07.20
7.	Адаптация и реализация существующих нейросетевых архитектур для решения задачи распознавания людей.	17-19.07.20
8.	Обучение построенной модели на размеченных данных, адаптация и применение модели к исходным данным компании.	20.07.20
9.	Передача результатов работы алгоритмов в компанию.	21.07.20
10.	Оформление отчёта.	22.07.20

С заданием ознакомлен(а) \_\_\_\_\_



(подпись обучающегося)

**Руководитель практики от СПбГЭУ**

\_\_\_\_\_  
(Подпись)

Лебедева Людмила Николаевна  
(Расшифровка)

**Руководитель практики от профильной организации**

(заполняется в случае прохождения практики в профильной организации)

\_\_\_\_\_  
(Подпись)

М.П.

Юрецкий Алексей Владимирович  
(Расшифровка)

## Содержание

ВВЕДЕНИЕ .....	4
1. ПРАКТИКА В ООО «ПЛАЗ» .....	5
1.1 Общая информация о компании .....	5
1.2 Технологии и методы разработки, используемые в компании .....	5
1.3 Организация работы практикантов.....	6
2. ИНДИВИДУЛЬНОЕ ЗАДАНИЕ .....	7
2.1 Описание задачи .....	7
2.2 Описание данных.....	7
2.3 Математическая постановка задачи и этапы решения .....	9
2.4 Нейросетевые методы решения. ....	10
2.4.1 Архитектура U-Net .....	12
2.4.2 Результаты.....	15
2.5 Применение модели к исходным данным .....	16
2.5.1 Схема решения.....	16
2.5.2 Результаты.....	17
ЗАКЛЮЧЕНИЕ .....	19
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	20
ПРИЛОЖЕНИЕ А .....	21

## **ВВЕДЕНИЕ**

В соответствии с учебным планом, в период с 9 июля 2019 года по 22 июля 2019 года мною была пройдена производственная практика в ООО «ПЛАЗ».

Целью производственной практики является изучение разновидностей профессиональной деятельности, формирование практических навыков и умений, которые потребуются при дальнейшей работе.

Прохождение практики является актуальным, так как студентам необходимо понимать, как строится работа в фирмах. Получение первого профессионального опыта, приобретение навыков, использование полученных в учебном заведении знаний – все это только подтверждает актуальность производственной практики.

Основным результатом производственной практики является выполнение индивидуального задания – прикладной задачи компании, которая является актуальной на данный момент.

## **1. ПРАКТИКА В ООО «ПЛАЗ»**

### **1.1 Общая информация о компании**

ООО «ПЛАЗ» является частью компании Геоскан и отвечает за разработку алгоритмов компьютерного зрения и оптимизации полётов беспилотных летательных аппаратов.

Геоскан - российская группа компаний, занимающаяся разработкой и производством беспилотных летательных аппаратов (БПЛА), а также разработкой программного обеспечения для фотограмметрической обработки данных и трехмерной визуализации.

Геоскан специализируется на выпуске коммерческих БПЛА двух типов: самолетного и мультироторного. Одной из главных особенностей беспилотников является то, что все комплексы создаются на собственном производстве «Геоскана». В производственном процессе используются лишь некоторые компоненты стороннего производства, а компоненты - собственной разработки позволяют серьезно снизить себестоимость продуктов.

Завод по производству РЭА и беспилотных комплексов, конструкторское бюро проектирования РЭА и БПЛА и отдел разработки программного обеспечения расположены на площади 5500 кв. м в Санкт-Петербурге. Группа имеет филиалы в Москве, Белгороде и Сургуте.

“Геоскан” на данный момент осуществляет поставки в 12 стран мира. Уникальными преимуществами техники “Геоскан” перед другими являются рекордные показатели продолжительности и протяженности полета. Качество беспилотников ГК “Геоскан” позволяет конкурировать и превосходить большинство зарубежных аналогов.

### **1.2 Технологии и методы разработки, используемые в компании**

Для задач распознавания образов на аэросъемке компания использует современные методы компьютерного зрения. В компании широко используются нейросетевые методы компьютерного зрения, а так же современные методы обработки изображений.

Для задач оптимизации полётов беспилотных летательных аппаратов компания использует современные методы оптимизации и математического программирования, а так же языки программирования, которые обеспечивают быстроту и качество решения поставленных задач.

### **1.3 Организация работы практикантов.**

В первый день производственной практики прошёл ряд лекций, в которых студентам, проходящим практику в компании, рассказывалась история компании, достижения, нюансы работы и техника безопасности.

Обучение проходило по следующим темам:

1. Вводное слово о компании
2. Инструктаж ОУП
3. Инструктаж по организации Адаптационного обучения
4. Вводный инструктаж ООТПБ
5. Инструктаж СМК
6. Инструктаж ГОИВИ
7. Инструктаж СТА
8. Первичный инструктаж по охране труда на рабочем месте
9. Первичный инструктаж по пожарной безопасности
10. Знакомство практиканта с коллективом подразделения

Практика в компании проходила дистанционно, поэтому основной деятельностью практикантов являлось решение индивидуальных заданий, предоставленных компанией.

Работу над индивидуальными заданиями курировали сотрудники компании.



## **2. ИНДИВИДУЛЬНОЕ ЗАДАНИЕ**

### **2.1 Описание задачи**

Одной из задач компании является поиск людей, пропавших без вести. Для поиска этих людей требуется отправить беспилотные летательные аппараты для съемки местности, где может находиться пропавший человек.

В результате такой съемки появляется достаточно большое количество фото и видео материала, который необходимо просмотреть. На это требуется большое количество времени и человеческих усилий.

Возникает потребность разработать алгоритм, который автоматически определяет, присутствует ли человек на данной фотографии.

Компания предоставила данные аэросъемки с беспилотных летательных аппаратов. Моей задачей является разработка алгоритмов распознавания людей на данных снимках.

### **2.2 Описание данных**

Компания предоставила данные аэросъемки с беспилотных летательных аппаратов. Данные выглядят следующим образом.



Рисунок 1 - Данные аэросъемки

Всего имеется 431 снимок в разрешении (5472, 3648) пикселей. На снимках имеются объекты следующих типов:

1. Человек
2. Дерево
3. Дорога
4. Автомобиль

Стоит отметить, что только на 25 снимках, предоставленных компанией, присутствуют люди.

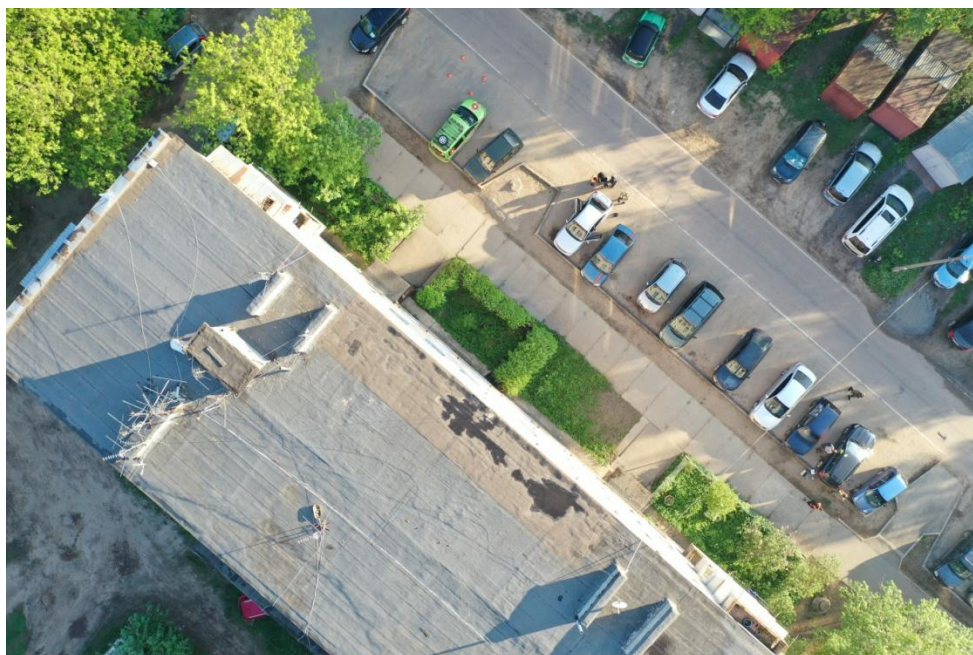


Рисунок 2 - Снимок с людьми

Данные, предоставленные компанией, не являются размеченными. Это значит, что не пописано, на какой фотографии присутствует человек. Тем более не указано, в какой области он находится, что является необходимым условием для задачи распознавания образов.

Возникает потребность в дополнительных данных для решения задачи сегментации и распознавания образов. Данные должны быть размеченными, на них должны присутствовать объекты всех типов, перечисленных для исходных данных компании.

Для решения задачи выбран набор данных *semantic – drone – dataset<sub>1</sub>*, ссылка на который указана в источниках. В данных имеется 400 изображений аэросъемки с беспилотных летательных аппаратов. Данные выглядят следующим образом.

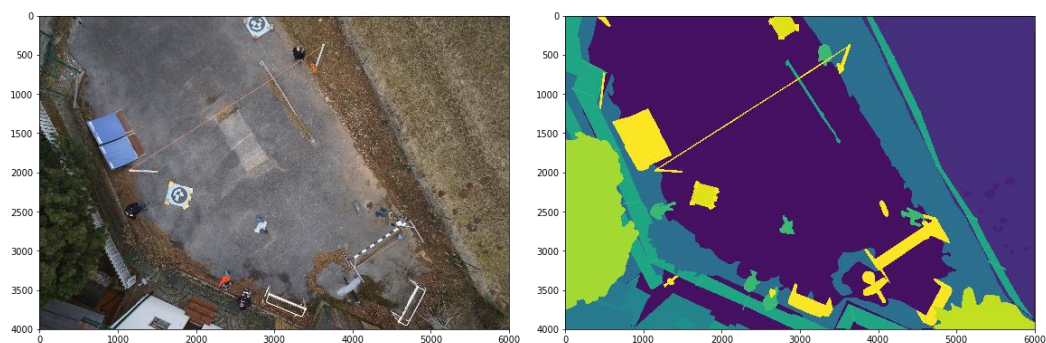


Рисунок 3 - Размеченные данные аэросъемки

В данных присутствуют объекты 23 типов: дерево, трава, другая растительность, грязь, гравий, камни, вода, мощеная территория, бассейн, человек, собака, автомобиль, велосипед, крыша, стена, забор, забор, окно, дверь, препятствие.

Каждому объекту соответствует свой цвет.

### 2.3 Математическая постановка задачи и этапы решения

Задача формулируется следующим образом. Определить, есть ли на снимке предметы, которые похожи на человека. Это задача бинарной классификации изображений.

Стоит отметить, что в задаче поиска людей крайне важно найти все изображения, на которых может быть человек. Этот факт влияет на выбор целевой функции.

Рассмотрим матрицу ошибок.

	Predicted class	
	Class = Yes	Class = No
Actual Class	Class = Yes	Class = No
	True Positive	False Negative
	False Positive	True Negative

Рисунок 4 - Матрица ошибок

Задача поиска людей предполагает максимизировать  $TP$  — *true positive* и минимизировать  $FN$  — *false negative*.

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (1)$$

$$recall = \frac{TP}{TP + FN} \quad (2)$$

Считается, что изображения, на которых присутствуют люди, относятся к классу *positive*. Это означает, что оптимальной метрикой в данной задаче будет *recall*.

Решение задачи бинарной классификации для изображений такого размера предполагается строить в несколько этапов. На изображениях присутствуют объекты других классов, их необходимо отличать от людей. Этапы решения задачи будут выглядеть следующим образом.

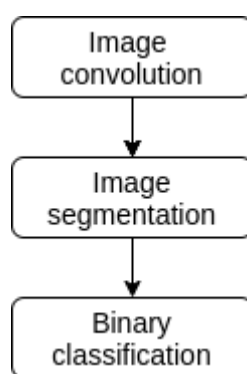


Рисунок 5 - Этапы решения

Первым делом будут рассмотрены методы сжатия изображений для приемлемых размеров. Далее будут рассмотрены методы решения задачи распознавания образов и сегментации изображений, после чего обученная модель сегментации будет применяться к исходным данным для задачи бинарной классификации.

Целевые функции для задачи сегментации:

$$multiclass\ accuracy = \sum_{i=1}^{n-clases} \frac{TP_i + TN_i}{TP_i + FP_i + TN_i + FN_i} \quad (3)$$

Для каждого размеченного изображения и для каждого класса будет *accuracy*.

## 2.4 Нейросетевые методы решения.

Для каждого размеченного изображения и для каждого класса будет *accuracy*.

Задачей данного раздела является разработка и применение архитектуры сверточных нейронных сетей для задачи распознавания образов.

Основные архитектуры, применяемые для решения данной задачи, представлены на рисунке.

<b>model_name</b>	<b>Base Model</b>	<b>Segmentation Model</b>
fcn_8	Vanilla CNN	FCN8
fcn_32	Vanilla CNN	FCN8
fcn_8_vgg	VGG 16	FCN8
fcn_32_vgg	VGG 16	FCN32
fcn_8_resnet50	Resnet-50	FCN32
fcn_32_resnet50	Resnet-50	FCN32
fcn_8_mobilenet	MobileNet	FCN32
fcn_32_mobilenet	MobileNet	FCN32
pspnet	Vanilla CNN	PSPNet
vgg_pspnet	VGG 16	PSPNet
resnet50_pspnet	Resnet-50	PSPNet
unet_mini	Vanilla Mini CNN	U-Net
unet	Vanilla CNN	U-Net
vgg_unet	VGG 16	U-Net
resnet50_unet	Resnet-50	U-Net
mobilenet_unet	MobileNet	U-Net
segnet	Vanilla CNN	Segnet
vgg_segnet	VGG 16	Segnet
resnet50_segnet	Resnet-50	Segnet
mobilenet_segnet	MobileNet	Segnet

Рисунок 6 - Основные архитектуры нейронных сетей для задачи сегментации

По результатам работы, а так же в силу специфики задачи, в качестве архитектуры была выбрана модель *U – Net* на базе *VGG 16*.

Архитектура модели состоит из последовательных блоков. Входными данными является изображение. Необходимо предсказать сегментацию

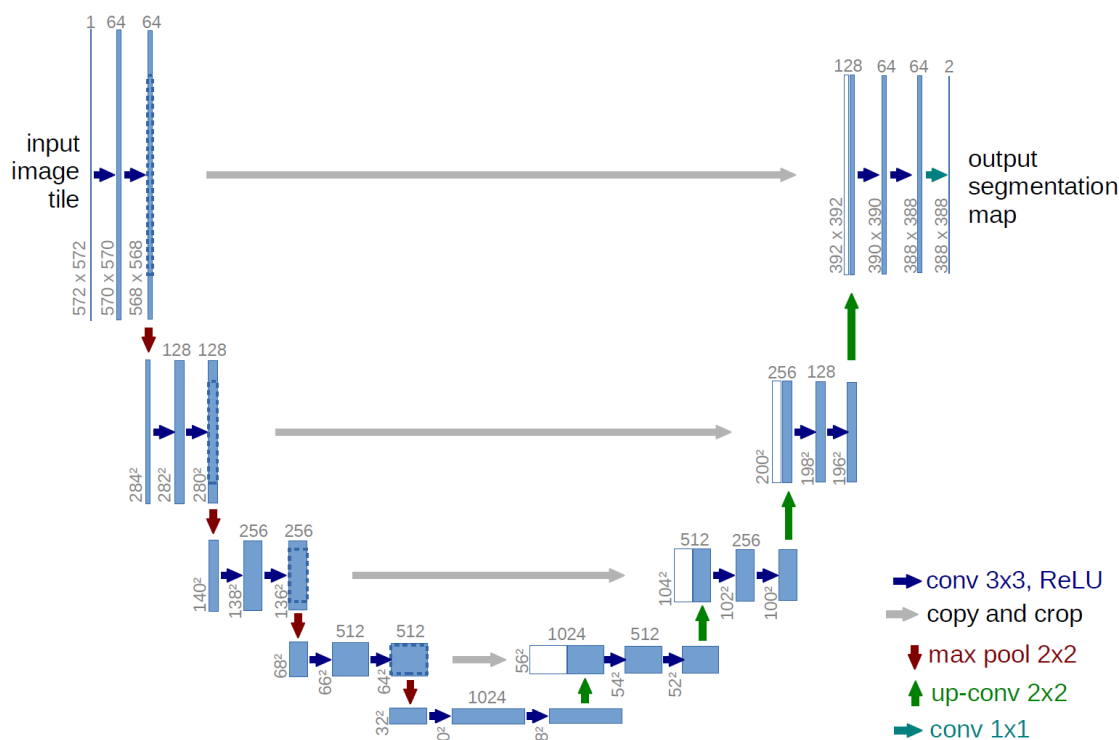


данного изображения, то есть предоставить изображение, на котором предсказанные области классов будут выделены.

### 2.4.1 Архитектура U-Net

UNET был разработан Olaf Ronneberger et al. для биомедицинской сегментации изображения. Архитектура содержит два пути. Первый путь - это путь сжатия (также называемый кодером), который используется для захвата контекста в изображении. Кодер - это просто традиционный набор сверток и максимальных уровней пулинга. Второй путь - это симметричный путь расширения (также называемый декодером), который используется для обеспечения точной локализации с использованием транспонированных сверток. Таким образом, это сквозная полностью сверточная сеть (FCN), то есть она содержит только сверточные слои и не содержит какого-либо плотного слоя, из-за чего она может принимать изображение любого размера.

Модель состоит из последовательных операций свертки и пулинга, между блоками модели присутствуют слои нормализации.



### Рисунок 7 - Архитектура U-Net

В результате работы были определены оптимальные параметры модели, такие как метод оптимизации, количество блоков и скорость обучения.

Модель обеспечивает значение целевой функции в задаче сегментации изображений для тестовой выборки порядка 72%. Это означает, что в 72% случаев мы можем распознать объект, если он принадлежит к одному из классов, описанных ранее.

Другие параметры модели.

$$optimizer = Adam \quad (4)$$

$$learning\ rate = 0.01 \quad (5)$$

$$accuracy = 0.72 \quad (6)$$

Модель обеспечивает значение целевой функции в задаче сегментации изображений для тестовой выборки порядка 72%. Это означает, что в 72% случаев мы можем распознать объект, если он принадлежит к одному из классов, описанных ранее.

Полученная модель решает задачу сегментации для изображений *semantic – drone – dataset<sub>1</sub>*.

Модель можно рассматривать в два этапа, как показано на рисунке. Первый этап отвечает за сжатие изображения с помощью операторов свёртки и пулинга. Когда достигнута необходимая размерность, решается задача сегментации для сжатого изображения. После этого необходимо восстановить исходный размер изображения, за это отвечает второй этап модели. С помощью обратных преобразований свёртки размер изображения увеличивается.

Архитектура второго этапа представлена ниже.

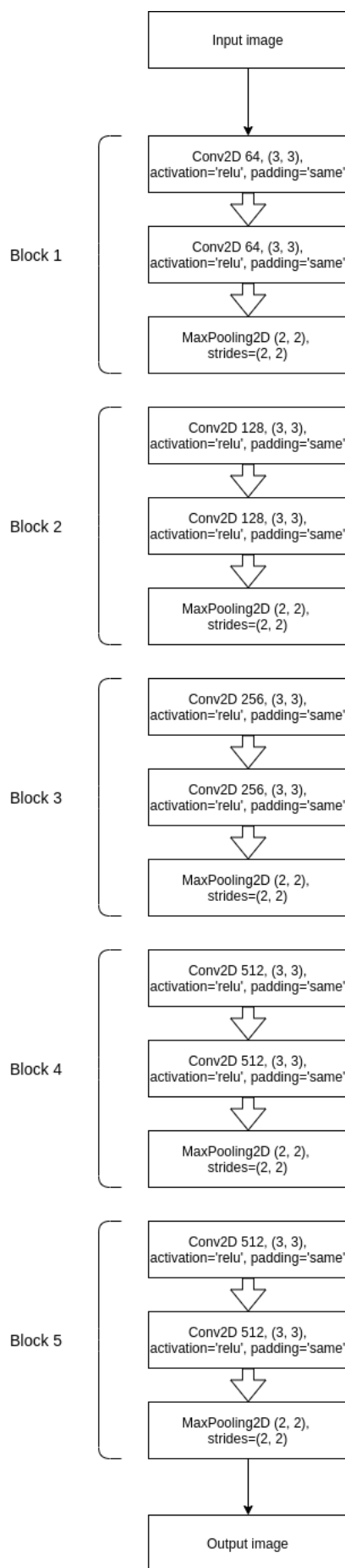


Рисунок 8 - Архитектура модели восстановления изображения



## 2.4.2 Результаты

Модель качественно предсказывает сегментацию изображений для тестовой выборки. Основным достоинством данной архитектуры является то, что благодаря большому количеству свёрточных слоёв изображение возможно сжать до необходимой размерности. Это ускоряет обучение модели с незначительными потерями информации.

Таким образом в модель представляется возможным загружать изображения исходной размерности.

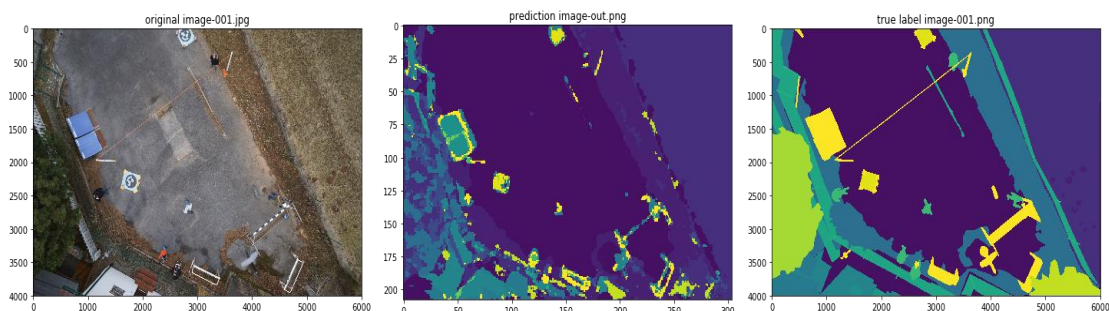


Рисунок 9 - Пример сегментации 1

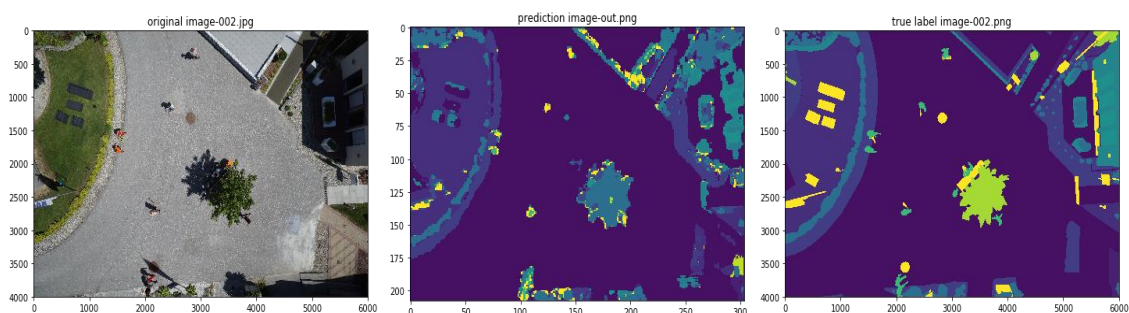


Рисунок 10 - Пример сегментации 2

Примеры демонстрируют, что объекты типа «человек» достаточно хорошо распознаются моделью. Так же отчётливо видны объекты типа «препятствие», «автомобиль» и «дерево».

Качество модели:

$$multiclass\ accuracy = 0.72, people\ recall = 0.65. \quad (7)$$

Стоит отметить, что исходная задача заключается в распознавании людей на аэросъемке. Это означает, что данная модель является вспомогательной, далее необходимо применить её к исходным данным компании «Геоскан» в задаче бинарной классификации.

## **2.5 Применение модели к исходным данным**

### **2.5.1 Схема решения**

В результате работы была получена модель для решения задачи распознавания образов на изображениях с беспилотных летательных аппаратов. Модель распознаёт объекты следующих типов.

1. дерево
2. трава
3. другая растительность
4. грязь
5. гравий
6. камни
7. вода
8. мощеная территория
9. бассейн
10. человек
11. собака
12. автомобиль
13. велосипед
14. крыша
15. стена
16. забор
17. окно
18. дверь
19. препятствие

Исходные данные не являются размеченными для задачи сегментации, представляется возможным самостоятельно разметить их только для задачи бинарной классификации.

Соответственно, исходная модель может предсказывать сегментацию. Если на сегментированном изображении присутствует объект типа

«человек», а так же исходное изображение имеет метку «1», то предсказание считается верным. Аналогично, если изображение имеет метку «0», а на сегментированном изображении нет объектов типа человек.

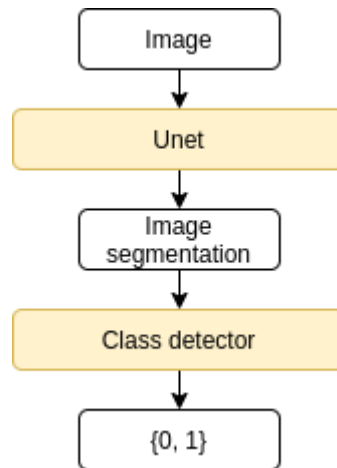


Рисунок 11 - Схема решения

### 2.5.2 Результаты

Рассмотрим примеры применения модели  $U - Net$ , обученной на наборе данных  $semantic - drone - dataset_1$ , к исходным данным. Для исходных данных будут посчитаны целевые метрики, определенные в математической постановке задаче, а так же будут рассмотрены сегментированные изображения, полученные из исходных данных.

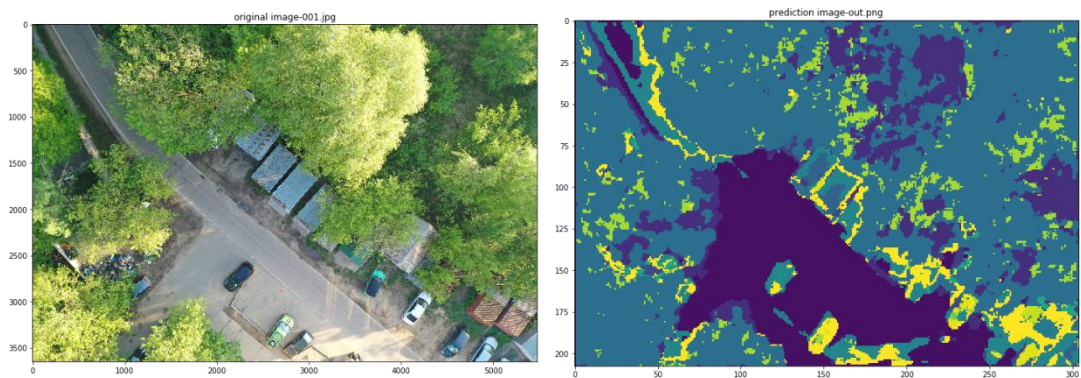


Рисунок 12 - Пример сегментации исходных данных 1

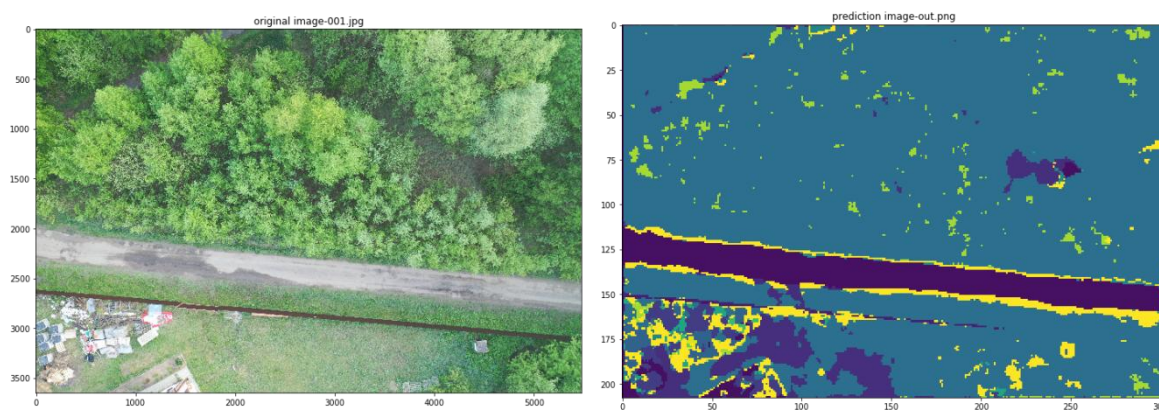


Рисунок 13 - Пример сегментации исходных данных 2

Исходные данные не являются размеченными для задачи сегментации, представляется возможным самостоятельно разметить их только для задачи бинарной классификации.

Данные примеры показывают, что объекты типа «дерево» и «дорога» распознаются хорошо, в то время как объекты типа «человек» распознаются хуже.

Посчитаем целевые метрики для задачи бинарной классификации:

$$accuracy\ people = 0.83 \quad (8)_{\text{ы}}$$

$$recall\ people = 0.57 \quad (9)$$

Такие результаты можно объяснить неоднородностью данных. Действительно, из исходных данных, предоставленных компанией, только в шести процентах имеются люди.

В таком случае, если выбирать *accuracy* в качестве метрики, то лучший результат будет давать наивный прогноз. Тем не менее, модель может позволить сегментировать исходные данные, что облегчит задачу в будущем.

## **ЗАКЛЮЧЕНИЕ**

По результатам производственной практики была проделана следующая работа.

1. Исходные данные компании были размечены для задачи бинарной классификации.
2. Был найден набор данных схожей природы, который является размеченным для задачи сегментации изображений. Набор данных находится в открытом доступе.
3. Для вспомогательного набора данных была решена задача сегментации изображений.
4. Для исходных данных была решена задача бинарной классификации изображений.

В результате прохождения производственной практики были изучены современные методы обработки изображений и распознавания образов. Были реализованы и использованы модификации популярных архитектур свёрточных нейронных сетей.

Стоит отметить, что решение задачи без вспомогательного набора данных не представляется возможным. Как было отмечено ранее, необходимые объекты присутствуют только в шести процентах изображений.

Планируется продолжение работы над задачей компании, в случае предоставления компанией более полных данных со снимками людей с беспилотных летательных аппаратов.

Полученная модель может применяться не только для распознавания людей, но и для распознавания других классов объектов, присутствующих в вспомогательном наборе данных.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Положение об организации и проведении практики обучающихся, осваивающих основные профессиональные образовательные программы высшего образования в ФГБОУ ВО «Санкт-Петербургский государственный экономический университет»
2. Положение о структуре и оформлении письменных работ обучающимися по программам среднего профессионального и высшего образования в ФГБОУ ВО «Санкт-Петербургский государственный экономический университет»
3. Jain, V., Murray, J. F., Roth, F. Supervised learning of image restoration with convolutional networks: In Computer Vision, 2007. ICCV 2007. IEEE 11<sup>th</sup> International Conference on, pages 1-8: 2007.
4. Semantic drone dataset [Электронный ресурс] / Режим доступа: <https://www.kaggle.com/bulentsiyah/semantic-drone-dataset>, свободный. – Загл.с экрана. – Яз. англ. Дата обращения: 14.07.2020.
5. Распознавание образов [Электронный ресурс] / Режим доступа: <https://habr.com/ru/company/ods/blog/431512/>, свободный. – Загл.с экрана. – Яз. англ. Дата обращения: 11.07.2020.
6. Модели сегментации [Электронный ресурс] / Режим доступа: [https://keras.io/api/metrics/segmentation\\_metrics/](https://keras.io/api/metrics/segmentation_metrics/), свободный. – Загл.с экрана. – Яз. англ. Дата обращения: 17.07.2020.

# ПРИЛОЖЕНИЕ А

## Код на языке Python

### 1. Загрузка библиотек

```
import os
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '3'

import keras
from keras.models import *
from keras.layers import *

from types import MethodType
import random
import six
import json
from tqdm import tqdm
import cv2
import numpy as np
import itertools
```

### 2. Определение гиперпараметров

```
IMAGE_ORDERING_CHANNELS_FIRST = "channels_first"
IMAGE_ORDERING_CHANNELS_LAST = "channels_last"
# Default IMAGE_ORDERING = channels_last
IMAGE_ORDERING = IMAGE_ORDERING_CHANNELS_LAST

if IMAGE_ORDERING == 'channels_first':
    MERGE_AXIS = 1
elif IMAGE_ORDERING == 'channels_last':
    MERGE_AXIS = -1

if IMAGE_ORDERING == 'channels_first':
    pretrained_url = "https://github.com/fchollet/deep-learning-models/" \
        "releases/download/v0.1/" \
        "vgg16_weights_th_dim_ordering_th_kernels_notop.h5"
elif IMAGE_ORDERING == 'channels_last':
    pretrained_url = "https://github.com/fchollet/deep-learning-models/" \
        "releases/download/v0.1/" \
        "vgg16_weights_tf_dim_ordering_tf_kernels_notop.h5"

class_colors = [(random.randint(0, 255), random.randint(
    0, 255), random.randint(0, 255)) for _ in range(5000)]
```

### 3. Вспомогательные функции

```
def get_colored_segmentation_image( seg_arr , n_classes , colors=class_colors ):
    output_height = seg_arr.shape[0]
    output_width = seg_arr.shape[1]

    seg_img = np.zeros((output_height, output_width, 3))

    for c in range(n_classes):
        seg_img[:, :, 0] += ((seg_arr[:, :] == c)*(colors[c][0])).astype('uint8')
        seg_img[:, :, 1] += ((seg_arr[:, :] == c)*(colors[c][1])).astype('uint8')
        seg_img[:, :, 2] += ((seg_arr[:, :] == c)*(colors[c][2])).astype('uint8')

    return seg_img

def get_vgg_encoder(input_height=224, input_width=224, pretrained='imagenet'):

    assert input_height % 32 == 0
    assert input_width % 32 == 0

    if IMAGE_ORDERING == 'channels_first':
        img_input = Input(shape=(3, input_height, input_width))
    elif IMAGE_ORDERING == 'channels_last':
        img_input = Input(shape=(input_height, input_width, 3))

    x = Conv2D(64, (3, 3), activation='relu', padding='same',
               name='block1_conv1', data_format=IMAGE_ORDERING)(img_input)
    x = Conv2D(64, (3, 3), activation='relu', padding='same',
               name='block1_conv2', data_format=IMAGE_ORDERING)(x)
    x = MaxPooling2D((2, 2), strides=(2, 2), name='block1_pool',
                    data_format=IMAGE_ORDERING)(x)
    f1 = x
    # Block 2
    x = Conv2D(128, (3, 3), activation='relu', padding='same',
               name='block2_conv1', data_format=IMAGE_ORDERING)(x)
    x = Conv2D(128, (3, 3), activation='relu', padding='same',
               name='block2_conv2', data_format=IMAGE_ORDERING)(x)
    x = MaxPooling2D((2, 2), strides=(2, 2), name='block2_pool',
                    data_format=IMAGE_ORDERING)(x)
    f2 = x

    # Block 3
    x = Conv2D(256, (3, 3), activation='relu', padding='same',
```



```

        name='block3_conv1', data_format=IMAGE_ORDERING)(x)
x = Conv2D(256, (3, 3), activation='relu', padding='same',
        name='block3_conv2', data_format=IMAGE_ORDERING)(x)
x = Conv2D(256, (3, 3), activation='relu', padding='same',
        name='block3_conv3', data_format=IMAGE_ORDERING)(x)
x = MaxPooling2D((2, 2), strides=(2, 2), name='block3_pool',
        data_format=IMAGE_ORDERING)(x)
f3 = x

# Block 4
x = Conv2D(512, (3, 3), activation='relu', padding='same',
        name='block4_conv1', data_format=IMAGE_ORDERING)(x)
x = Conv2D(512, (3, 3), activation='relu', padding='same',
        name='block4_conv2', data_format=IMAGE_ORDERING)(x)
x = Conv2D(512, (3, 3), activation='relu', padding='same',
        name='block4_conv3', data_format=IMAGE_ORDERING)(x)
x = MaxPooling2D((2, 2), strides=(2, 2), name='block4_pool',
        data_format=IMAGE_ORDERING)(x)
f4 = x

# Block 5
x = Conv2D(512, (3, 3), activation='relu', padding='same',
        name='block5_conv1', data_format=IMAGE_ORDERING)(x)
x = Conv2D(512, (3, 3), activation='relu', padding='same',
        name='block5_conv2', data_format=IMAGE_ORDERING)(x)
x = Conv2D(512, (3, 3), activation='relu', padding='same',
        name='block5_conv3', data_format=IMAGE_ORDERING)(x)
x = MaxPooling2D((2, 2), strides=(2, 2), name='block5_pool',
        data_format=IMAGE_ORDERING)(x)
f5 = x

if pretrained == 'imagenet':
    VGG_Weights_path = keras.utils.get_file(pretrained_url.split("/")[-1],
pretrained_url)
    Model(img_input, x).load_weights(VGG_Weights_path)

return img_input, [f1, f2, f3, f4, f5]

```

#### 4. Создание модели

```

def _unet(n_classes, encoder, l1_skip_conn=True, input_height=416,
        input_width=608):

    img_input, levels = encoder(

```

```

    input_height=input_height, input_width=input_width)
[f1, f2, f3, f4, f5] = levels

o = f4

o = (ZeroPadding2D((1, 1), data_format=IMAGE_ORDERING))(o)
o = (Conv2D(512, (3, 3), padding='valid',
data_format=IMAGE_ORDERING))(o)
o = (BatchNormalization())(o)

o = (UpSampling2D((2, 2), data_format=IMAGE_ORDERING))(o)
o = (concatenate([o, f3], axis=MERGE_AXIS))
o = (ZeroPadding2D((1, 1), data_format=IMAGE_ORDERING))(o)
o = (Conv2D(256, (3, 3), padding='valid',
data_format=IMAGE_ORDERING))(o)
o = (BatchNormalization())(o)

o = (UpSampling2D((2, 2), data_format=IMAGE_ORDERING))(o)
o = (concatenate([o, f2], axis=MERGE_AXIS))
o = (ZeroPadding2D((1, 1), data_format=IMAGE_ORDERING))(o)
o = (Conv2D(128, (3, 3), padding='valid',
data_format=IMAGE_ORDERING))(o)
o = (BatchNormalization())(o)

o = (UpSampling2D((2, 2), data_format=IMAGE_ORDERING))(o)

if l1_skip_conn:
    o = (concatenate([o, f1], axis=MERGE_AXIS))

o = (ZeroPadding2D((1, 1), data_format=IMAGE_ORDERING))(o)
o = (Conv2D(64, (3, 3), padding='valid',
data_format=IMAGE_ORDERING))(o)
o = (BatchNormalization())(o)

o = Conv2D(n_classes, (3, 3),
padding='same', data_format=IMAGE_ORDERING)(o)

model = get_segmentation_model(img_input, o)

return model

def vgg_unet(n_classes, input_height=416, input_width=608, encoder_level=3):

```

```

    model = _unet(n_classes, get_vgg_encoder, input_height=input_height,
input_width=input_width)
    model.model_name = "vgg_unet"
    return model

```

## 5. Обучение модели

n\_classes = 23 # Aerial Semantic Segmentation Drone Dataset tree, gras, other vegetation, dirt, gravel, rocks, water, paved area, pool, person, dog, car, bicycle, roof, wall, fence, fence-pole, window, door, obstacle

```

model = vgg_unet(n_classes=n_classes, input_height=416, input_width=608)
model_from_name = {}
model_from_name["vgg_unet"] = vgg_unet

```

```

model.train(
    train_images = "/kaggle/input/semantic-drone-
dataset/semantic_drone_dataset/original_images/",
    train_annotations = "/kaggle/input/semantic-drone-
dataset/semantic_drone_dataset/label_images_semantic/",
    checkpoints_path = "vgg_unet" , epochs=epochs
)

```