**Федеральное агентство связи**

**Ордена Трудового Красного Знамени**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский технический университет связи и информатики»**

Кафедра «Информатики»

**Отчет по практической работе**

на тему:

«Обработка изображений и распознавание образов»

Выполнил: студент группы

БВТ1901

Перевозчиков С. В.

Руководитель:

Мосева М. С.

Москва 2020

Оглавление

[1. Постановка задачи и анализ предметной области 3](#_Toc52043695)

# Постановка задачи и анализ предметной области

Требуется создать нейросеть для детектирования люков на дорожном полотне.

Системы распознавания различных дефектов дорожного полотна и, в частности, люков могут применяться в различных службах, следящих за состоянием дорог. Для выполнения данной задачи наиболее удобным решением может служить применение сверточной нейронной сети, преимуществом которых является слабая чувствительность к искажениям входного сигнала, из-за чего они часто применяются в задачах компьютерного зрения.

Сверточные нейронные сети (CNN) – разновидность нейронных сетей с алгоритмом «глубокого обучения», которая, принимая на входе некоторое изображение, может обращать внимание на области, а не только единичные элементы и, таким образом, распознавать объекты на изображении. В данном случае, получая на входе изображение, нейросеть должна распознать на нем люк, если он есть и выдать новое изображение с выделенным на нем люком или маску с расположением люка.

# Инструменты разработки

Для выполнения данной задачи использованы следующие инструменты:

1) Python 3.7.5 – язык программирования, на котором написаны нейронная сеть и алгоритмы подготовки данных для обучения

2) Дополнительные модули для Python: TensorFlow 2.3.0, Keras 2.4.3, opencv-python 4.4.0, NumPy 1.18.5 – требуются для работы нейросети и предварительной обработки изображений

3)Visual Studio 2019 – среда разработки для работы с файлами на языке Python и аннотациями в формате Json

4)Jupyter Notebook – удобный инструментарий для обучения нейронной сети

5)VGG Image Annotator – ПО, необходимое для аннотации изображений

# Обзор архитектур нейронных сетей

При поиске подходящей для выполнения задачи нейронной сети были рассмотрены разновидности нейронных сетей: полносвязные нейронные сети (FNN), рекуррентные нейронные сети (RNN) и сверточные нейронные сети (CNN).

**Полносвязной нейронной сетью прямого распространения** (Fully Connected Feed-Forward Neural Networks или FNN) называется нейронная сеть, в которой входные данные проходят через слои нейронов в одну сторону. Такие нейронные сети могут использоваться для классификации изображений, однако требуют большое количество данных для обучения.

**Рекуррентной нейронной сетью (**Recurrent Neural Networks или RNN) называется нейронная сеть, в которой входная информация может «циркулировать» по скрытым слоям, отправляясь более поздних нейронов на более ранние. Рекуррентные нейронные сети обладают дополнительной памятью за счет повторной передачи нейронной сети информации, но являются довольно сложными в обучении.

**Сверточной нейронной сетью (**Convolutional Neural Networks или CNN) называется нейронная сеть обладающая нейронами, реализующими свертку за счет сверточных слоев и уменьшение разрешения за счет слоев объединения (Pooling-слоев), и полносвязными слоями на выходе. Сверточные нейронные сети применяются при решении задач классификации, сегментации и распознавания.

Для выполнения поставленной задачи было решено использовать модель U-Net, способную работать с небольшим количеством изображений и сравнительно быстро обучаться за счет наличия у сети свойств как сверточной, так и рекуррентной нейронной сети.

# Сбор и разметка датасета для обучения и тестирования

Для распознавания люков на изображениях было сделано и размечено порядка 150 фотографий с одним или несколькими люками, из которых в силу ограничений памяти устройства, на котором производилось обучение, было использовано 30.

Для каждой исходной фотографии была создана маска с выделенным люком с помощью инструментов VIA и OpenCV.

Собранный датасет подвергся аугментации с помощью модуля ImageDataGenerator в Keras, после чего количество изображений и соответствующих им масок увеличилось до 20 тысяч.

# Обучение модели и ее тестирование

Код по обучению и тестированию модели предствлен на github в файлах trainTrapdoors.ipynb и testTrapdoors.ipynb соответственно.

Модель была обучена, но при тестировании показала неудовлетворительные результаты, не распознав люк ни на одном тестовом изображении.

# Заключение

Была изучена работа с инструментами по разметке изображений, синтаксис языка Python и некоторые библиотеки по работе с изображениями, рассмотрены архитектуры нейронных сетей и изучена нейронная сеть U-Net. При обучении модели возникли трудности, вызванные возможно недостаточным размером данных для обучения или низкой точностью их аннотации.