

Учреждение образования
“Белорусский государственный университет
информатики и радиоэлектроники”
Кафедра информатики

Отчёт по проекту:
«Детектирование ключевых точек рук»

Выполнил: Фисько Дмитрий Владимирович
магистрант кафедры информатики
группа № 956241

Проверил: Ассистент, магистр технических наук
Стержанов Максим Валерьевич

Минск 2019

Содержание

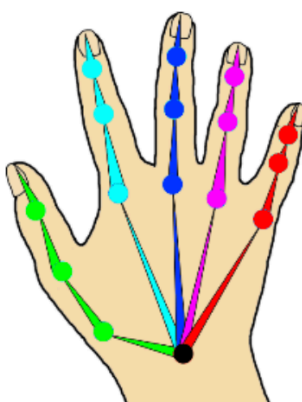
1.	Постановка задачи.....	3
2.	Процесс реализации задачи.....	3
3.	Хранение данных.....	4
4.	Модель.....	5
5.	Метрики качества	7
6.	Результаты:	9
7.	Список использованных источников.....	10

1. Постановка задачи

Необходимо создать и обучить модель для поиска ключевые точки рук на фотографиях. Фотографии рук поступают на вход модели в предобработанном виде и содержат обрезанное по области рук изображение. Пример входящих изображений:



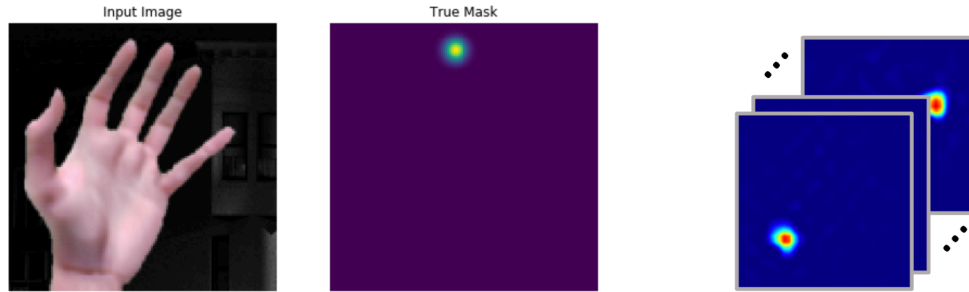
Необходимо в результате работы модели получить ключевые точки рук по входящему изображению. Всего точек в обучающей выборке 21. Пример результата работы модели.



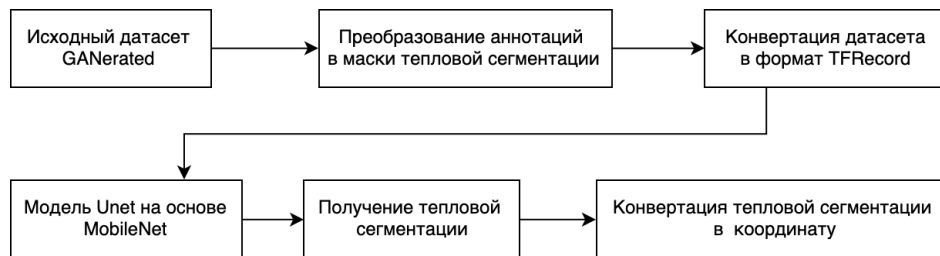
2. Процесс реализации задачи

Данные для обучения были взяты из датасета GANerated Hands. Каждый элемент данного датасета включает в себя файл с обрезанным изображением руки и соответствующую аннотацию ключевых точек рук.

В качестве модели были выбраны нейронная сетевая архитектура Unet. Чтобы появилась возможность использовать Unet в качестве модели для детекции ключевых точек. Каждая ключевая точка из исходного датасета приводилась к маске с точкой нормального распределения. Данная операция производилась с каждой точкой из аннотации. Используя данное преобразование, появилась возможность превратить задачу по детекции точек в соответствующую задачу сегментации.



Для проектирования нейросетевой модели и её последующего обучения был использован TensorFlow 2.0. В качестве Encoder части модели нейронной сети Unet был выбран MobileNet. В качестве Decoder нейронной сети были взяты слои сформированные через последовательную upscaling свёртку. Размер входного слоя 160x160. Потенциально данный выбор архитектуры позволит запускать нейронные сети на мобильном устройстве. Общая схема работы:



3. Хранение данных

Датасет GANerated Hands хранится в виде обычной файловой структуры, в которой есть JPEG картинка, а рядом с картинкой с похожим названием располагаются аннотации для ключевых точек.

```

0001_color_composed.png
0001_crop_params.txt
0001_joint2D.txt
0001_joint_pos.txt
0001_joint_pos_global.txt
  
```

Минусом данного подхода является его проблемы с производительностью. Каждое чтение файла изображения с диска приводило к операции декодирования картинки из JPEG, и перемещение курсора диска на отступ, с которым располагается картинка в файловой системе. Также нам нужно преобразовывать исходные аннотации в маску. Чтобы не выполнять данные операции лишь единожды при создании обучающей выборки исходные данные было решено перекодировать в protobuf формат называемый TFRecords.

Последовательное чтение данных для обучения нейронной сети из TFRecords не позволило скорости чтения данных стать узким местом для производительности процесса обучения нейросетевой модели.

В связи с ограниченностью исходных вычислительных ресурсов для обучения нейронной сети, был использованы не все элементы обучающей выборки, были взяты только первые 100 тысяч изображений из всего 700 тысяч доступных.

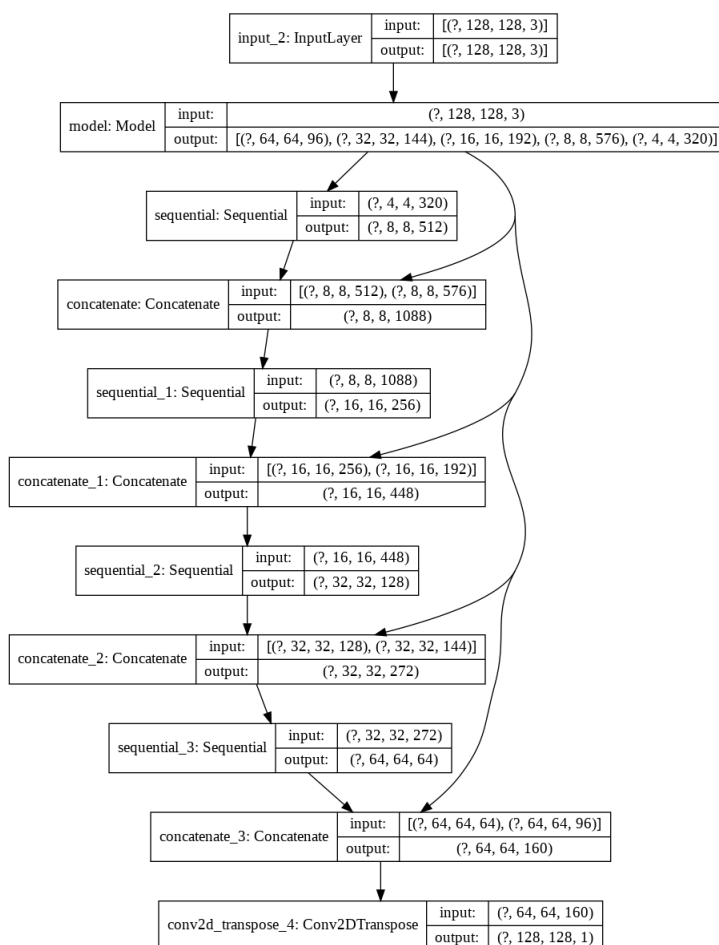
4. Модель

Часть кода по построению и обучению нейронной сети было взято из примера TensorFlow Examples. Большинство работы было проделано в процессе обработке и загрузке датасета. Исходный датасет был сконвертирован из обычных файлов в формат TFRecord. Это ускорило загрузку данных при обучении модели, устранив тем самым узкое место в скорости обучения.

Также была проделана работа по модифицированию исходной модели из примера. Исходная модель была ориентирована на создание классифицирующей сегментационной маски, поэтому использовался softmax в функции потерь. В модифицированной версии изменилась целевая функция модели, теперь она должна создавать сегментирующую тепловую карту, поэтому функция потерь была изменена на MSE.

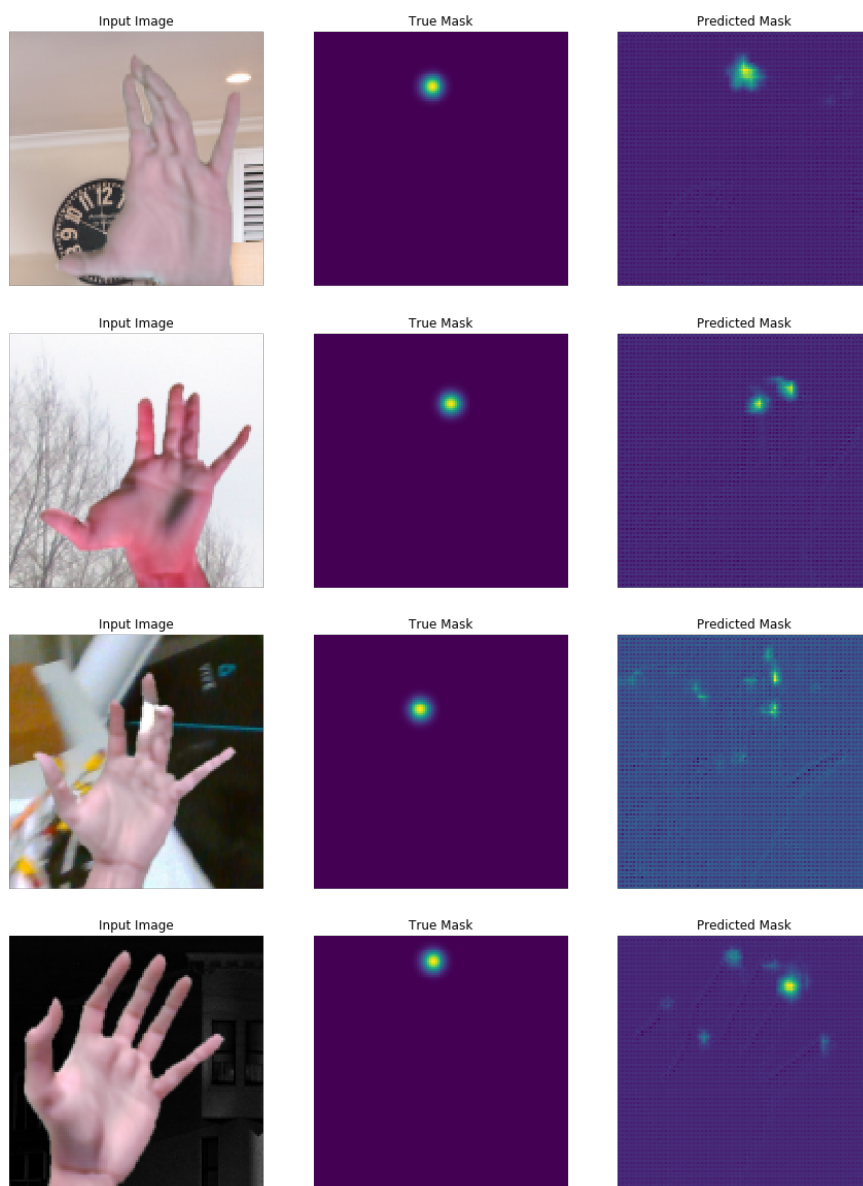
В качестве нейросетевой модели была выбрана Unet, с предобученной моделью MobileNet из пакета Keras в качестве Encoder для извлечения признаков. В качестве Decoder части была использована стандартная архитектура Unet с последовательным upscaling слоёв.

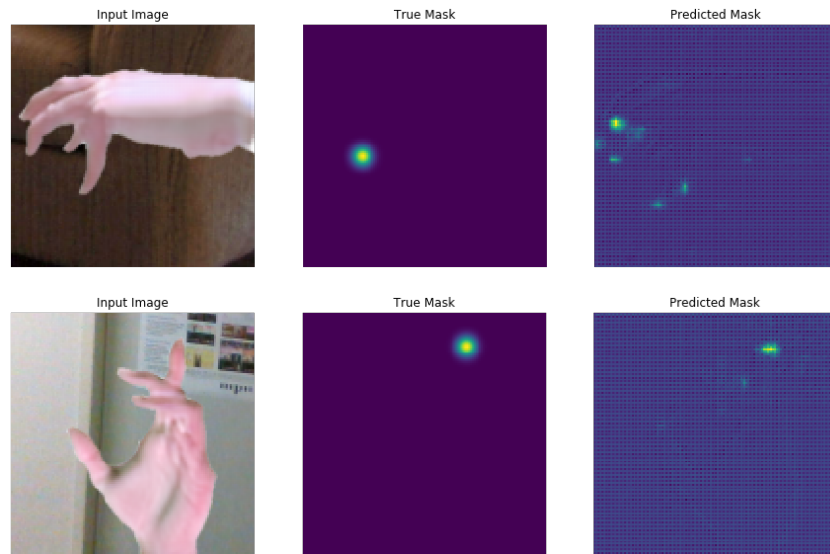
Итого получилась следующая архитектура нейронной сети:



Данная модель была в дальнейшем обучена на 90000 изображениях, и провалидирована на 10000 изображениях. В качестве функции потерь использовался квадрат попиксельной разницы между исходной маской для обучения и результатом модели. Обучение производилось в течении 20 эпох.

В итоге были получены следующие результаты для распознавания указательного пальца:





Как видно из результатов, обученная модель не всегда хорошо справляется со сложными тестовыми случаями. Данная проблема могла быть решена увеличением количества данных в тестовой выборке.

5. Метрики качества

Процесс обучения нейронной сети изображён ниже, всего было сделано 20 интерации над датасетом из 90 тысяч изображений. В итоге финальная метрика MSE была равна 1.1993×10^{-4} для обучающей выборке и 0.0023 для тестовой.

```
Train for 695 steps, validate for 15 steps
Epoch 1/20
319s 458ms/step - loss: 2.8777e-04 - val_loss: 0.0022
Epoch 2/20
319s 459ms/step - loss: 2.4053e-04 - val_loss: 0.0022
Epoch 3/20
319s 459ms/step - loss: 2.1979e-04 - val_loss: 0.0021
Epoch 4/20
318s 458ms/step - loss: 1.9479e-04 - val_loss: 0.0023
Epoch 5/20
319s 459ms/step - loss: 1.7418e-04 - val_loss: 0.0025
Epoch 6/20
319s 459ms/step - loss: 1.6475e-04 - val_loss: 0.0023
Epoch 7/20
319s 459ms/step - loss: 1.5843e-04 - val_loss: 0.0022
Epoch 8/20
319s 459ms/step - loss: 1.5231e-04 - val_loss: 0.0023
Epoch 9/20
319s 458ms/step - loss: 1.5660e-04 - val_loss: 0.0025
Epoch 10/20
318s 458ms/step - loss: 1.4389e-04 - val_loss: 0.0025
Epoch 11/20
319s 458ms/step - loss: 1.3622e-04 - val_loss: 0.0021
Epoch 12/20
319s 458ms/step - loss: 1.3599e-04 - val_loss: 0.0021
```

Epoch 13/20
319s 458ms/step - loss: 1.3422e-04 - val_loss: 0.0021
Epoch 14/20
319s 460ms/step - loss: 1.3552e-04 - val_loss: 0.0021
Epoch 15/20
319s 459ms/step - loss: 1.2018e-04 - val_loss: 0.0022
Epoch 16/20
319s 459ms/step - loss: 1.2932e-04 - val_loss: 0.0022
Epoch 17/20
317s 457ms/step - loss: 1.2129e-04 - val_loss: 0.0023
Epoch 18/20
318s 458ms/step - loss: 1.2134e-04 - val_loss: 0.0021
Epoch 19/20
318s 458ms/step - loss: 1.1965e-04 - val_loss: 0.0021
Epoch 20/20
319s 458ms/step - loss: 1.1993e-04 - val_loss: 0.0023

6. Результаты:

Была успешно спроектирована и обучена модель для детектирования одной ключевой точки на изображении руки. Исходная задача по детектированию ключевых точек руки была переформулирована в задачу сегментации, что позволило получить хорошие показатели на целевых метриках качества. Детектирование всех точек реализовано не было, но в ходе реализации проекта был получен опыт, позволяющий легко обобщить предоставленное решение на большее количество ключевых точек.

В процессе реализации проекта был получен опыт подготовки и очистке большого объёма данных. Из исходного датасета были убраны вхождения данных, которые были не реалистичными, тем самым улучшив сосредоточение модели на выделении наиболее важных признаков. Размер обработанного датасета составлял 30Gb. Из-за большого объёма данных, в ходе реализации алгоритмов по конвертации, отдельное внимание уделялось алгоритмам параллельной обработки данных. Предобработка датасета осуществлялась с эффективным использованием ресурсных мощностей. При этом в ходе обучения модели использовалась аугментация данных. В режиме реального времени, происходило видоизменение элементов датасета, не дающее нейронной сети переобучиться на данных для обучения.

Также был получен опыт проектирования модели и её обучения. Была изучена нейросетевая архитектура Unet. В исходной архитектуру модели Unet была добавлена предобученная модель MobileNet v2 для выделения признаков. В ходе реализации проекта был получен опыт использования фреймворка Tensorflow 2.0 совместно с библиотекой Keras.

7. Список использованных источников

- [1] Mpi-inf.mpg.de: [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://handtracker.mpi-inf.mpg.de/projects/GANeratedHands/GANeratedDataset.htm>. – Дата доступа: 20.11.2019.
- [2] Wikipedia.org: [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://en.wikipedia.org/wiki/U-Net>. – Дата доступа: 23.11.2019
- [3] CS229.stanford.edu: [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://cs229.stanford.edu/proj2014/Rahul%20Makhijani,%20Saleh%20Samaneh,%20Megh%20Mehta,%20Collaborative%20Filtering%20Recommender%20Systems.pdf>. – Дата доступа: 23.11.2019
- [4] Tensorflow.org: [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://www.tensorflow.org/tutorials/images/segmentation>. – Дата доступа: 25.11.2019
- [5] Github.com: [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://imgaug.readthedocs.io/en/latest/>. – Дата доступа: 25.11.201
- [6] Dlcourse.ai: [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://dlcourse.ai/> 6. – Дата доступа: 26.11.2019