

Семантический анализ фотографий с помощью глубоких нейронных сетей

Выпускная квалификационная работы

02.03.02 - Фундаментальная информатика и информационные технологии

Выполнил:

студент 4 курса Ивахненко Дмитрий Игоревич

Научный руководитель:

к. ф.-м. н., ст.преп. М. В. Юрушкин

24 июня 2020 г.

Институт ММиКН им. И.И. Воровича, Южный Федеральный Университет

В данной работе были поставлены следующие задачи:

- Проектирование архитектуры глубокой сверточной сети для задачи сегментации

- Подготовка сопровождающего кода для расширения возможностей обучения сети

- Реализация возможности полуавтоматической разметки данных

Пример задачи сегментации



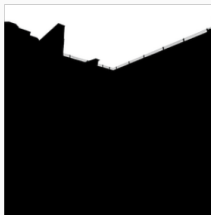
1. Результат работы сети



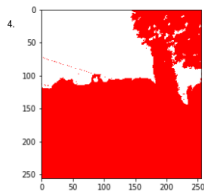
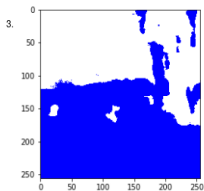
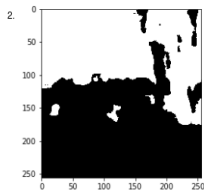
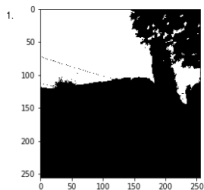
2. Исходное изображение



3. Коррекция результата

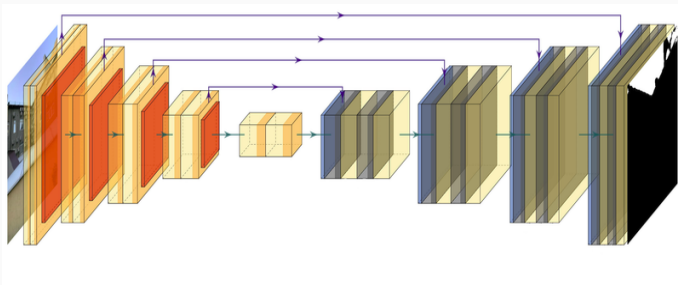


4. Размеченная маска



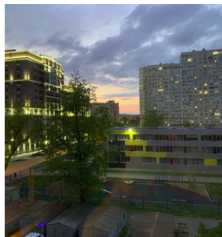
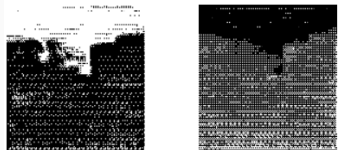
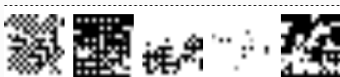
Визуализация метода сравнения площадей. 1. - истинная маска, 2. - предсказанная маска, 3. - пересечение, 4. - объединение

Общий вид FCN



Энкодер и декодер части сети. Позволяет выделить признаки, а затем генерализировать их для классификации.

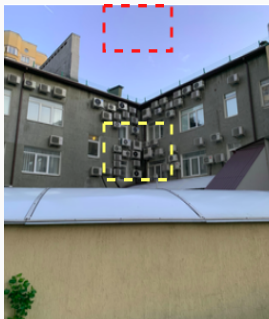
Выходы декодер-части



Нормализация по пакету



1. $\mu_B = \frac{1}{m} \sum_{i=1}^m x_i$
2. $\sigma_B^2 = \frac{1}{m} \sum_{i=1}^m (x_i - \mu_B)^2$
3. $\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$
4. $\text{BN}_{\gamma, \beta}(x_i) = \gamma \hat{x}_i + \beta$



```
Pre-softmax state; Size: torch.Size([2, 256, 256]),  
1. tensor([-9.2658939, 10.0882759], dtype=torch.float64),  
2. tensor([ 4.1472898, -4.4603243], dtype=torch.float64)
```

```
Post-softmax state; Size: torch.Size([2, 256, 256]),  
1. tensor([ 0.0000000, 1.0000000], dtype=torch.float64),  
2. tensor([ 0.9998173, 0.0001827], dtype=torch.float64)
```

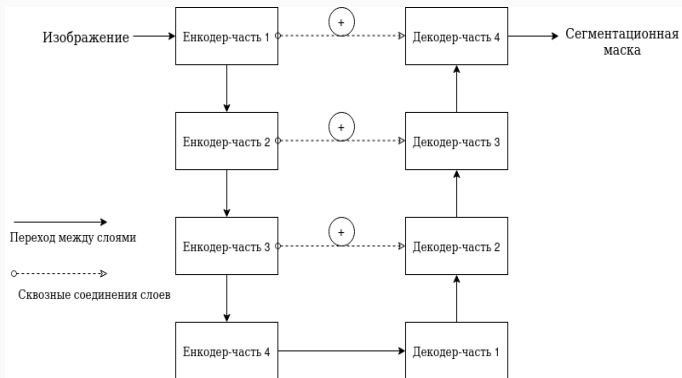
Формула:

$$\text{Softmax}(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}}$$

Свойства после применения

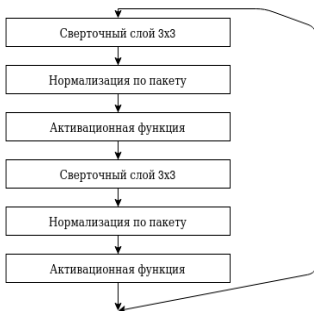
- $v_i \in [0, 1] \forall i \in [0, C]$
- $\sum_{i=0}^C v_i = 1$

Полная архитектура сети

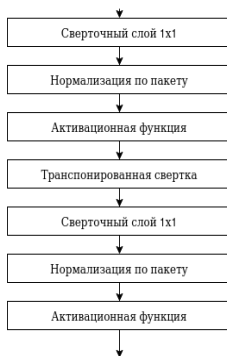


Обзор блоков

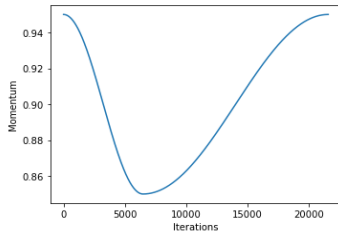
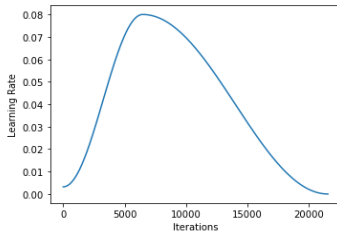
Енкодер блок по модели ResNet



Декодер блок по модели LinkNet



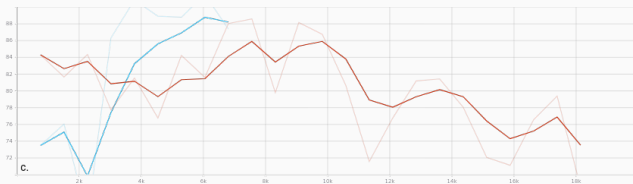
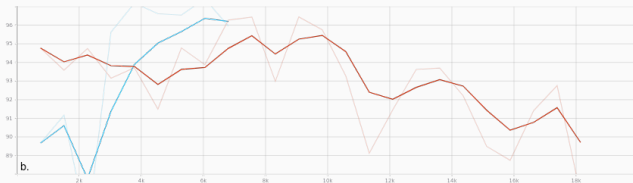
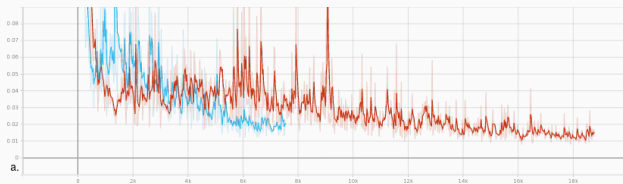
Оптимизация функции потерь



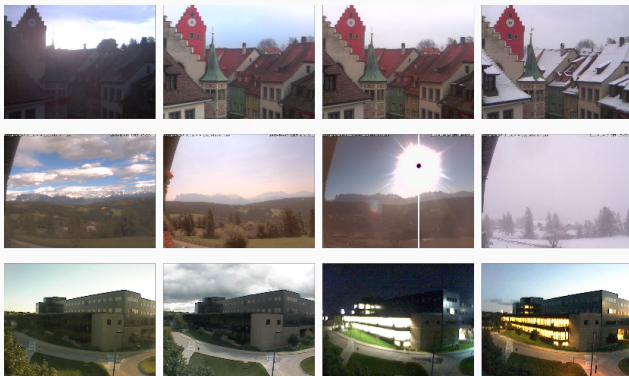
$$\omega_i = \omega_j - \left(\alpha \cdot \frac{\delta f(y, \tilde{y})}{\delta \omega_j} + \gamma \cdot \Delta \omega_k \right)$$

$$f(y, \tilde{y}) = -\frac{1}{n} \sum_{i=0}^n (\tilde{y}_i \cdot \log(y_i) + (1 - \tilde{y}_i) \cdot \log(1 - y_i))$$

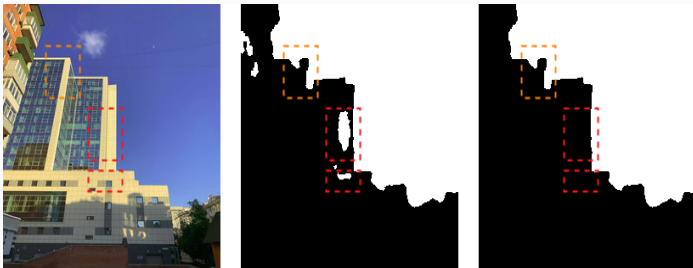
Сравнение процессов обучения



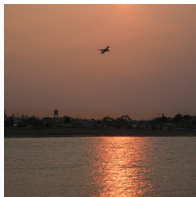
Примеры изображений SkyFinder



Коррекция результатов сети



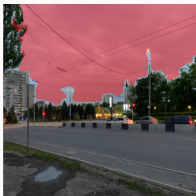
Синтетическая разметка



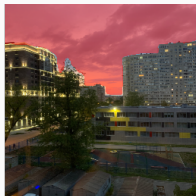
Результаты работы сети



1.



2.



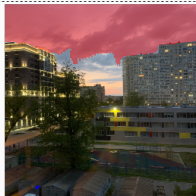
3.



4.



5.



6.

В рамках данной работы сделано:

- Спроектирована и реализована архитектура сети для решения задачи сегментации

- Реализована возможность обучения с использованием техники one cycle policy

- Реализована возможность синтетической разметки датасета

- Реализован загрузчик данных из нескольких источников с выбором количественного отношения в рамках пакета

Ссылка на репозиторий проекта:

<https://github.com/DmitIW/BachelorDiploma>