**SLIDE 2**

Семантическая сегментация это одна из задач решаемая компьютерным зрением.

Цель такой сегментации это отнести каждый пиксель к нужному классу. На картинке таких класса 3 – человек, велоспипед и фон. Но задача состоит не только в предсказании того или иного класса каждого пикселя, а еще и в выводе изображения, обычно того же разрешения, что и было на входе. Таким образом получаем классификацию изображений на уровне пикселей.

**SLIDE 3**

Про датасет, все очевидно

200 изображений поражений кожи: меланомы и родинки

**SLIDE 4**

Архитектура SegNet состоит из энкодера и декодера, каждый из которых состоит из сверточных блоков.

Сверточный блок энкодера устроен следующим образом: Сначала производятся несколько последовательных сверток, а затем max pooling, все вместе это называют downsampling

**SLIDE 5**

Операция max pooling’a уменьшает разрешение изображения в 2 раза (т.к. размер ядра 2х2). Идея этой операции – оставить только важные фичи изображения (т.е. пиксели с макс. значением). Делаем вывод: downsampling помогает модели понять ЧТО представлено на картинке, но модель теряет информацию о том ГДЕ это представлено.

**Вернуться к SLIDE 4**

Сверточный блок декодера состоит из upsampling (или max unpooling) и сверточных слоев, на выходе из последнего блока присутствует функция SoftMax (это используется в случае нескольких классов, у нас 2 класса).

**SLIDE 6**

Как уже было сказано, при downsampling мы теряем информацию о расположении пикселей, именно поэтому мы сохраняем индексы max pooling со слоев энкодера и пробрасываем их в слои декодера, это показано на картинке. Операция Upsampling увеличивает разрешение изображения в 2 раза (макс пуллинг уменьшал)

**SLIDE 7**

Вторая архитектура, которую мы использовали – U-Net. Она так же содержит блок энкодера и блок декодера, однако в отличие от SegNet здесь по-другому реализован Upsampling.

Блок энкодера, по сути, не отличается. Все те же сверточные слои и макс пуллинги.

В блоке декодера Upsampling реализован с помощью Transposed Convolutions (противоположность обычной операции свертки). С помощью таких сверток мы увеличиваем разрешение изображения, т.к. после энкодера мы имеем много карт признаков (feature maps), но все они малого разрешения, а мы хотим на выходе получить изображение с нормальным разрешением

В блоки декодера SegNet мы пробрасывали индексы макс пуллинга, а в U-Net мы этого не делаем, т.к. transposed convolutions имеют обучаемые параметры и помогают восстановить ту самую информацию о том, где находятся нужные пиксели в изображении.

Чтобы улучшить точность локализации таких пикселей авторы статьи использовали skip connections с помощью конкатенирования выхода после transposed convolutions и выхода с соответствующего слоя энкодера. Ну а затем после каждой конкатенации мы снова применяем обычные свертки чтобы модель обучалась выводить более точный результат.

**SLIDE 8**

Использована метрика IoU (intersection over union), обычно именно ее используют для оценки качества в задачах сегментации. Максимальное значение – 1.

**SLIDE 9**

В качестве функции потерь использован Tversky Loss (1 минус Tversky Index). Такая лосс функция используется при дисбалансе классов (в нашем случае фона (класс 0) больше, чем пикселей маски) и позволяет штрафовать сеть сильнее, если она предсказывает 0, когда на самом деле 1 (False Negative)

**SLIDE 13**

Видно, что скор у U-Net быстрее вышел на плато, по сравнению со скором SegNet.

Вообще говоря, размер датасета довольно маленький всего 200 изображений, 150 из которых были использованы для обучения, 50 для валидации и при увеличении его размера, результаты могли бы быть лучше.

LINKS:

orig paper segnet [https://arxiv.org/pdf/1511.00561.pdf](https://vk.com/away.php?utf=1&to=https%3A%2F%2Farxiv.org%2Fpdf%2F1511.00561.pdf)  
tvesky loss [https://towardsdatascience.com/dealing-with-class-imb..](https://vk.com/away.php?utf=1&to=https%3A%2F%2Ftowardsdatascience.com%2Fdealing-with-class-imbalanced-image-datasets-1cbd17de76b5)  
про юнет сегнет  
[https://towardsdatascience.com/understanding-semantic..](https://vk.com/away.php?utf=1&to=https%3A%2F%2Ftowardsdatascience.com%2Funderstanding-semantic-segmentation-with-unet-6be4f42d4b47)  
[https://towardsdatascience.com/review-segnet-semantic..](https://vk.com/away.php?utf=1&to=https%3A%2F%2Ftowardsdatascience.com%2Freview-segnet-semantic-segmentation-e66f2e30fb96)