

Алгоритмы сегментации COVID-19 на снимках компьютерной томографии

Курсовой проект

Минск, 2021

Содержание

Введение	3
1 Исходные данные	4
1.1 Медицинская терминология	4
1.2 Характеристики набора данных	5
2 Реализация модели	7
2.1 Обучение	7
2.2 Тестирование	7
2.3 Результаты	9
3 Улучшение результатов модели.	
Сравнение различных конфигураций	11
3.1 Аугментации	11
3.1.1 Реализация	11
3.1.2 Сравнение различных наборов аугментаций	12
3.2 Архитектуры	17
3.2.1 Обзор архитектур нейронных сетей для сегментации изображений	17
3.2.2 Сравнение результатов архитектур в данной задаче . . .	20
4 Заключение	22
Список использованных источников	23

Введение

Проблема так называемой сегментации изображений (image segmentation) стоит особенно остро в области медицины. Даже для квалифицированного специалиста определение патологий по различным снимкам (рентгеновским, томографическим и прочим) – нетривиальная задача. Так, согласно исследованиям, гепатоидная аденокарцинома, одна из опасных форм рака желудка, не диагностируется в 30% случаев, в том числе из-за неверного анализа снимков [1]. Поэтому изобретение технологий, помогающих определять патологии с точностью лучше человеческой, позволит вывести диагностику заболеваний на новый уровень и повысит качество здравоохранения в целом.

В наше время область машинного обучения стремительно развивается. Каждый год мы можем наблюдать новые инновации и достижения, в том числе в области распознавания изображений. Нейронные сети научились распознавать человеческие лица, виды животных и растений, особенности окружающего мира (вроде дорожных знаков в городе). Сегментация медицинских изображений, однако, имеет некоторые особенности, которые отличают ее от этих задач. Так, требуется более высокая точность, при этом данные (снимки пациентов и размеченные специалистами патологии), напротив, находятся в дефиците.

В условиях пандемии COVID-19 спрос на скорость и качество диагностики только вырос. Когда количество пациентов превышает возможности медицинского персонала, особенно важно найти способы уменьшить нагрузку на специалистов. Одним из таких способов является внедрение технологий в медицину.

Данная работа исследует вопрос сегментации патологий, вызываемых COVID-19 (симптом «матового стекла», лёгочная консолидация), на снимках компьютерной томографии легких. Рассмотрены способы улучшить эффективность обучаемых моделей в виде аугментаций, а также проведено сравнение различных архитектур нейронных сетей, предназначенных для сегментации медицинских изображений.

1. Исходные данные

Мы будем использовать набор данных «COVID-19 CT segmentetion dataset» [2], а также его предварительно обработанную версию из соответствующего соревнования на платформе Kaggle [3].

1.1 Медицинская терминология

Компьютерная томография (computed tomography). Метод исследования внутреннего строения предмета. Разные по плотности ткани по-разному влияют на рентгеновское излучение (например, могут сильно ослабить его). Эти различия измеряются и обрабатываются компьютером. В настоящее время КТ – один из основных методов исследования внутренностей человека.

КТ-сканеры представляют собой рентгеновскую трубу, внутри которой расположены различные детекторы. Снимок, полученный в результате одного оборота трубы, называется КТ-срезом. Существуют 1, 2, 4, 16, 100, 300-срезовые (-axial) томографы. Один срез представляет собой срез плоскости органа (толщиной в несколько мм).

Симптом «матового стекла» (ground-glass) [4]. Визуализируется при КТ легких как туманное, матовое снижение прозрачности лёгочной ткани с явным рисунком сосудов и бронхов на этом фоне. Данный симптом неспецифичен, т. е. его причиной могут быть разные болезни, поэтому при постановке диагноза лишь на нем не основываются.

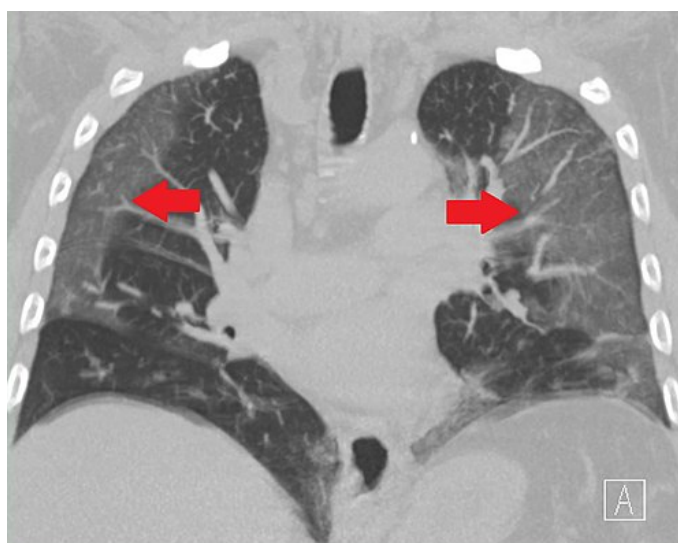


Рисунок 1.1 — Симптом «матового стекла»
на снимке больного COVID-19 пациента

Легочная консолидация (consolidation) [5]. Симптом, при котором легочная ткань уплотняется из-за того, что альвеолы, которые в нормальном состоянии наполнены воздухом, заполняются жидкостью (например, кровью или водой). Наличие этого симптома является необходимым условием для постановки диагноза пневмонии.

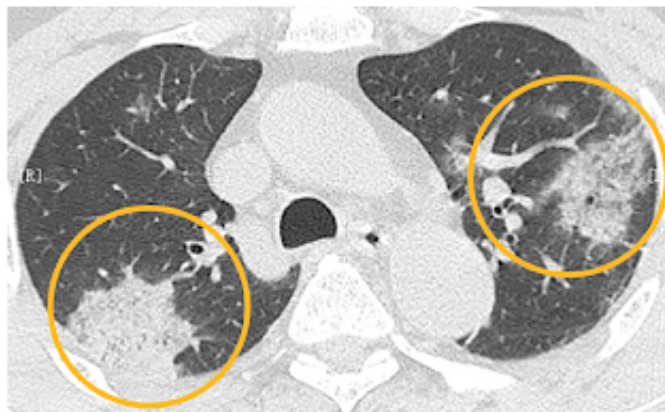


Рисунок 1.2 — Легочная консолидация на снимке больного COVID-19 пациента

1.2 Характеристики набора данных

«COVID-19 CT segmentetion dataset» состоит из двух частей:

- **MedSeg** – 100 односрезовых снимков более чем 40 больных COVID-19. Оригинальные снимки взяты из базы Итальянского Радиологического Общества [6]. Размер снимков был изменен до 512x512, сами они были сделаны черно-белыми. После этого яркость снимков была нормализована (т. е. теперь одинаковые ткани имеют приблизительно одинаковый оттенок на всех снимках).
- **Radiopaedia** – 829 снимков, из них 373 принадлежали больным COVID-19 [7]. Снимки были аналогично сделаны черно-белыми с нормализованной яркостью, но их размер не обязательно равен 512x512. Кроме того, данные снимки 9-срезовые, т. е. каждые девять снимков являются срезом одного и того же органа. Radiopaedia – ресурс, созданный профессиональными радиологами с целью предоставить свободный доступ ко всем имеющимся знаниям в области радиологии.

Снимки размечены вручную, с маркировкой матового стекла, легочной консолидации и плеврального выпота. Также известно, какие из снимков принадлежат больным.

Kaggle-версия набора данных имеет некоторые различия:

- Размер снимков в части «Radiopaedia» – 512x512.
- Формат данных – numpy-массивы, а не изображения.
- На снимках размечены: матовое стекло, легочная консолидация, легочная ткань, фон (не легкие).

Мы будем пользоваться kaggle-версией ввиду более удобного формата.

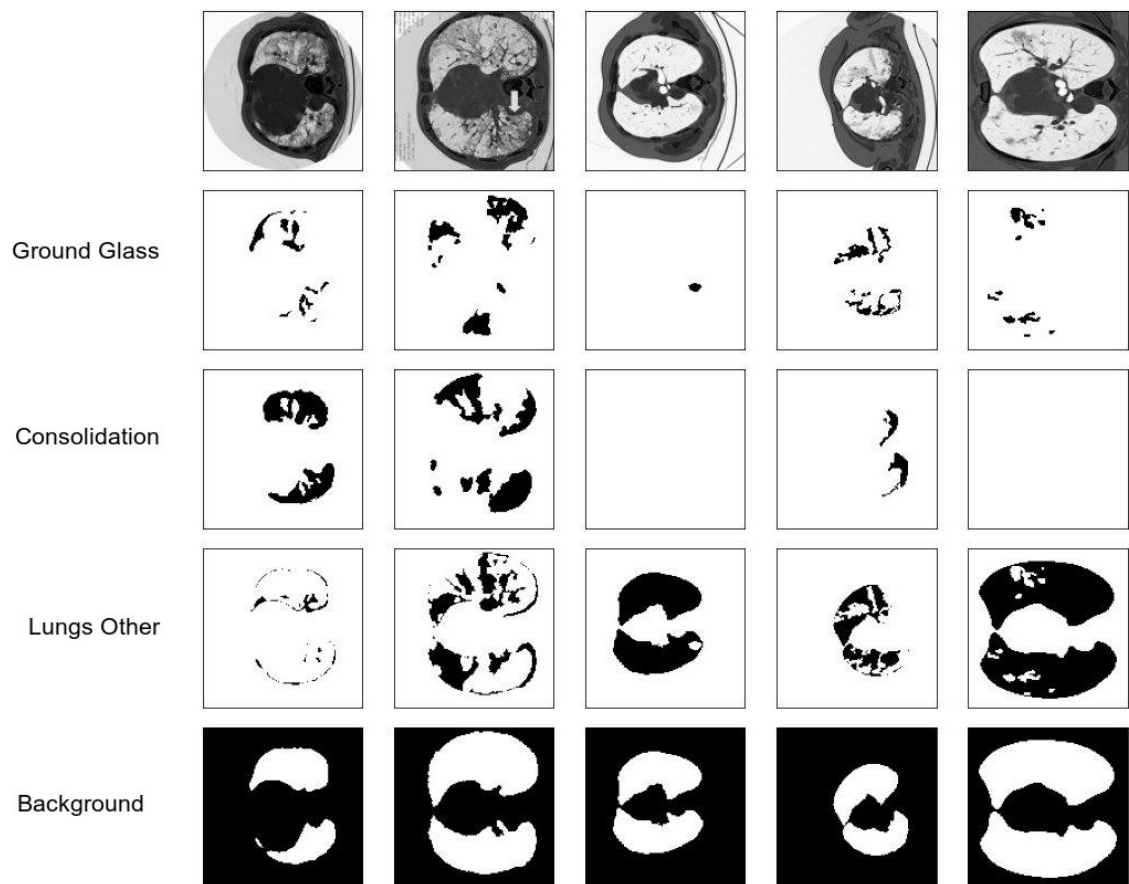


Рисунок 1.3 — Примеры изображений и их разметки

2. Реализация модели

Для обучения нейронных сетей будем пользоваться библиотекой PyTorch [8]. Реализации различных архитектур нейронных сетей возьмем из библиотеки `segmentation-models-pytorch` [9]. Для обучения модели реализуем классы `Dataset` и `Network`¹.

Класс `Dataset`. Представляет собой абстракцию набора данных и используется для удобной манипуляции ими, например, для разделения на порции («мини-батчи»).

Отметим, что мы уменьшили размер изображений с 512x512 до 256x256 для ускорения вычислений и уменьшения нагрузки на физические ресурсы, такие как оперативная память.

Класс `Network`. Представляет собой абстракцию нейросети. Его параметры – архитектура сети и некоторые необходимые для обучения метрики, такие как функция потерь (loss-функция). В самом базовом варианте мы пользуемся архитектурой Unet и кросс-энтропией в виде функции потерь.

2.1 Обучение

Обучение модели проводится в несколько эпох. В течение одной эпохи модель обрабатывает все данные по одному разу.

За одну эпоху для каждого мини-батча:

1. Вычисляется предсказание модели;
2. Вычисляется значение функции потерь для найденного предсказания;
3. При помощи функции-оптимизатора и алгоритма обратного распространения градиента (backpropagation) вычисляются новые веса модели.

Обучение модели мы проводим до сходимости, т. е. останавливаемся, когда значение функции потерь перестает уменьшаться.

2.2 Тестирование

Тестирование модели проводится на части набора данных, не участвовавшей в обучении. Таким образом мы можем видеть, как хорошо

¹Исходный код находится в Github-репозитории: <https://github.com/elizabethfeden/covid-tomography>.

модель поведет себя в реальных условиях. Эффективность модели оцениваем по следующим метрикам:

1. **Точность (accuracy)**. Число правильных ответов, деленное на общее их количество. Идеальная оценка равна 1, или 100%. Это простейшая оценка, которая не учитывает ни особенностей задачи сегментации изображений, ни особенностей самого набора данных, однако позволяет получить базовое представление об эффективности модели.
2. **IoU (Intersection over Union, пересечение на объединение, мера Жаккара)**. Метрика, специфическая для задачи сегментации изображений. Она вычисляется для конкретного класса сегментации i . Введем обозначения: A_{true}^i – множество пикселей, принадлежащих классу i ; A_{pred}^i – множество пикселей, принадлежащих классу i по предсказанию модели; $I_i = A_{pred}^i \cap A_{true}^i$; $U_i = A_{pred}^i \cup A_{true}^i$. Тогда оценка $IoU(i)$ определяется как

$$IoU(i) = \frac{|I_i|}{|U_i|}$$

Для идеальной модели $I_i = U_i \forall i$, поэтому идеальная оценка $IoU(i) = 1 \forall i$. Таким образом, метрика позволяет понять, насколько хорошо модель сегментирует конкретный класс.

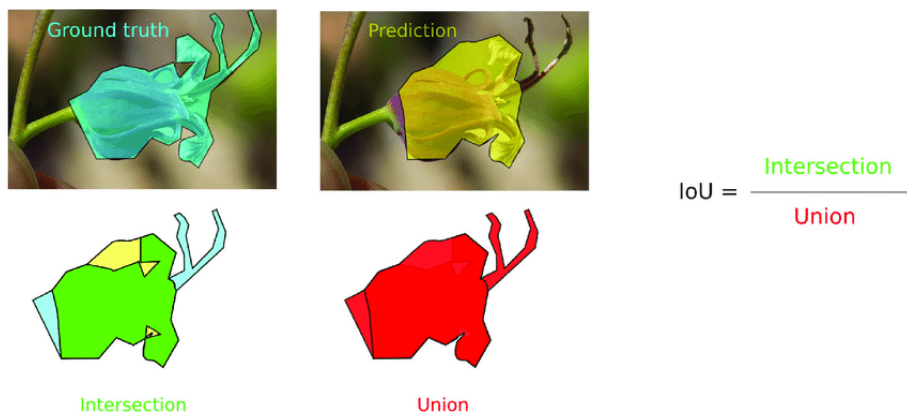


Рисунок 2.1 — Пример нахождения IoU

Существует также метрика $mIoU$ (среднее IoU), равная $\frac{1}{n} \sum_i IoU(i)$, где n – число классов сегментации. Однако мы будем обращать больше внимания именно на отдельные IoU , так как в данной задаче разные классы представляют для нас разный интерес; так, эффективность выделения областей консолидации приоритетнее эффективности выделения областей с фоном.

2.3 Результаты

Для начала мы обучили модель с базовыми параметрами на двух классах (фон и легкие) и на четырех классах. Были получены следующие результаты:

Epochs	50
Loss	3.7936
Accuracy	0.9960
Lungs IoU	0.9766
Background IoU	0.9953

Таблица 2.1 — Результаты тестирования модели, обученной на двух классах

Видим, что модель крайне эффективно отличает легкие от фона. Можно сказать, что с этой достаточно простой задачей обученная нами модель справляется также хорошо, как справился бы человек. Так как улучшение результатов модели для двух классов не представляет большого интереса, далее будем всегда работать с четырьмя классами.

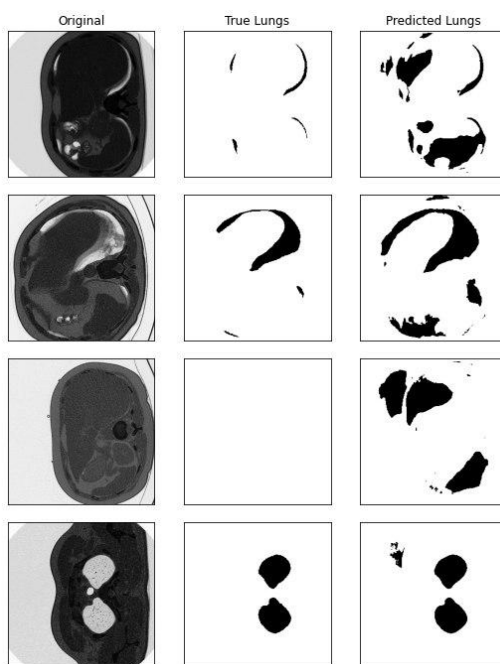


Рисунок 2.2 — Примеры предсказаний базовой модели (два класса)

Epochs	41
Loss	9.028
Accuracy	0.9891
Ground Glass IoU	0.6709
Consolidation IoU	0.5052
Lungs Other IoU	0.9407
Background IoU	0.9946

Таблица 2.2 — Результаты тестирования модели, обученной на четырех классах

Для четырех классов все не так однозначно; модель все еще показывает хорошие результаты на более простых классах (легкие и фон), что положительно влияет на метрику точности. Однако по покласовым IoU видно, что предсказания патологий очень далеки от идеала. Дальнейшие исследования посвятим повышению эффективности предсказания патологий.

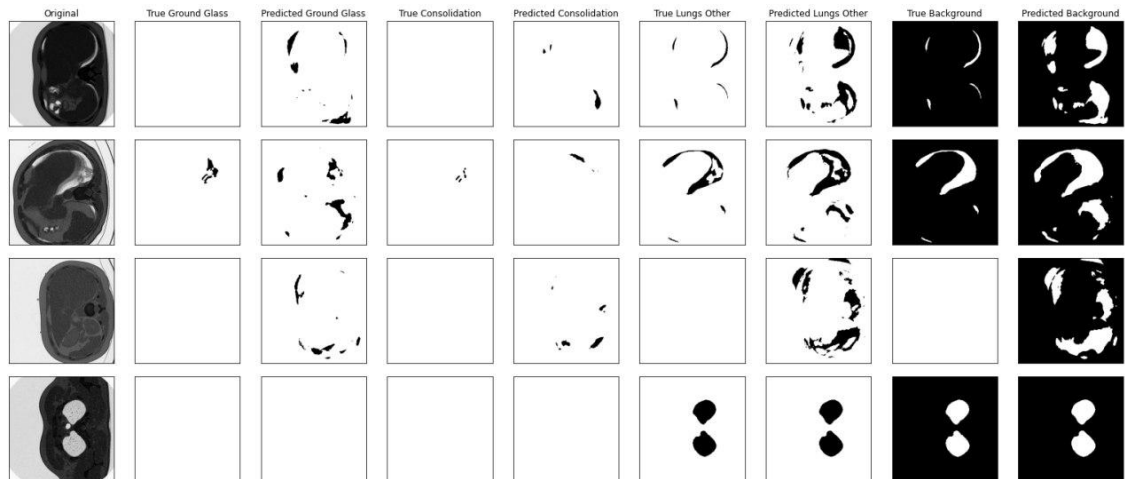


Рисунок 2.3 — Примеры предсказаний базовой модели (четыре класса)

3. Улучшение результатов модели. Сравнение различных конфигураций

3.1 Аугментации

Аугментации – произвольные изменения изображений (например, поворот на произвольное число градусов, отражение, добавление шума). На каждой эпохе обучения модели к каждому изображению применяется новая, произвольная, аугментация.

Использование аугментаций позволяет модели лучше обобщаться, так как она должна будет адаптироваться к изменениям изображения. Модель будет искать признаки, общие для всех вариаций одного изображения, а не заучивать паттерны на одних и тех же данных. Таким образом, аугментации также являются хорошим средством от переобучения моделей.

Аугментации также позволяют существенно уменьшить размер набора данных, требуемый для обучения нейронных сетей. Часто для обучения модели необходимы тысячи изображений, найти и разметить правильные ответы для которых – сложная задача. Особенно проблема небольших выборок актуальна для задач машинного обучения, связанных с медициной [10]: снимки пациентов в общем случае крайне сложно найти в открытом доступе, а в случае с сегментацией изображений также необходимо, чтобы их вручную разметил квалифицированный специалист.

3.1.1 Реализация

Для аугментации изображений воспользуемся библиотекой `albumentations` [11].

```
1 import albumentations as albu
2
3 train_transform = albu.Compose([
4     albu.VerticalFlip(p=0.5),
5     albu.OneOf([
6         albu.ShiftScaleRotate(rotate_limit=15, shift_limit=0.05,
7             scale_limit=0.05, p=0.5, border_mode=cv2.BORDER_REPLICATE),
8         albu.GridDistortion(p=0.5),
9     ], p=0.3),
10    albu.RandomBrightnessContrast(brightness_limit=0.1, contrast_limit=0.1,
11    p=0.5),
12 ])
```

Листинг 3.1: Пример создания аугментации при помощи `albumentations`

3.1.2 Сравнение различных наборов аугментаций

Выбор подходящего набора аугментаций – интересная задача. Необходимо учитывать распределение изначальных данных: если аугментированные данные принадлежат другому распределению, то модель будет показывать плохие результаты на тестирующей выборке, к которой аугментации не применяются. Рассмотрим различные наборы аугментаций и результаты модели, к которым они приводят.

Простейшие аугментации

Набор аугментаций состоит из композиции простейших аугментаций. Для начала мы рассмотрели наборы, состоящие из одной простейшей аугментации.

Вертикальное отражение (Vertical Flip, VFlip). Будем далее обозначать эту аугментацию как V^p , где p – вероятность, с которой она применяется. Такое отражение, примененное к изображению, меняет местами левое и правое легкое. В некоторых случаях это может привести к ухудшению работы модели – если, например, определенные патологии характерны лишь для правого легкого. Однако в нашем случае результат улучшился, поэтому можно сделать вывод об отсутствии таких специфичных патологий.

Epochs	59
Loss	9.0246
Accuracy	0.9896
Ground Glass IoU	0.6958
Consolidation IoU	0.4556
Lungs Other IoU	0.9435
Background IoU	0.9948

Таблица 3.1 — Результаты модели на данных, аугментированных $V^{0.5}$

Отметим, что горизонтальное отражение мы применять не будем, так как снимки легких не симметричны по этой оси.

Масштабирование, поворот, сдвиг (ShiftScaleRotate). Будем далее обозначать эту аугментацию как $S^p[rotate, shift, scale]$, где p – вероятность, с которой она применяется; $rotate$ – максимальное число градусов, на которое поворачивается изображение (т. е. изображение может быть повернуто на случайное число $\xi \sim R[0; rotate]$ градусов); $shift$ – максимальный размер сдвига изображения в произвольном направлении (в процентах от размера изображения); $scale$ – максимальный процент от размера изображения, на который оно увеличивается или уменьшается.

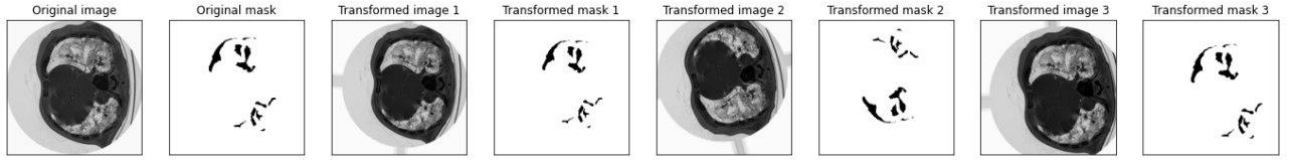


Рисунок 3.1 — Пример применения аугментации $V^{0.5}S^1[15, 0.1, 0.1]$ к одному изображению

Эта аугментация – одна из самых распространенных. В самом деле, она часто позволяет увеличить разнообразие изображений без влияния на распределение данных.

	$V^{0.5}S^1[15, 0.1, 0.1]$	$V^{0.5}S^1[15, 0.05, 0.05]$
Epochs	63	76
Loss	9.0484	9.0352
Accuracy	0.9896	0.9900
mIoU	0.7962	0.8063

Таблица 3.2 — Сравнение результатов модели на данных, аугментированных $V^{0.5}S^1$ с различными параметрами

Мы также пробовали аугментировать $V^{0.5}S^1[20, 0.15, 0.15]$, но получили несравнимо худшие результаты, т. е. качество модели на несколько десятых уступило либо качеству базовой модели без аугментаций, либо качеству наилучшей модели с данным набором аугментаций, но другими параметрами. Такие модели рассматривать далее не имеет смысла, однако полезно знать, какие параметры показывают худшие результаты.

Яркость и контрастность (RandomBrightnessContrast). Будем далее обозначать эту аугментацию как $B^p[brightness, contrast]$, где p – вероятность, с которой она применяется; *brightness* – максимальный процент, на который изменяется яркость (при *brightness* = 1 получим полностью белое изображение); *contrast* – аналогично, но для контрастности изображения.



Рисунок 3.2 — Пример применения аугментации $B^1[0.3, 0.3]$ к одному изображению

Эта аугментация хорошо подходит именно для нашего случая: снимки, сделанные на разных томографах, обладают разными яркостью и контрастом, при этом наличие патологии не зависит от общей яркости изображения, что позволит модели хорошо находить определенные паттерны.

В результате экспериментов мы получили, что $B^1[0.1, 0.1]$ показывает наилучший результат, а $B^1[0.05, 0.05]$ и $B^1[0.2, 0.2]$ несравнимо хуже.

Растяжение по сетке (GridDistortion). Будем далее обозначать эту аугментацию как $G^p[distortion]$, где p – вероятность, с которой она применяется; $distortion$ обозначает максимальную интенсивность растяжения (в процентах).

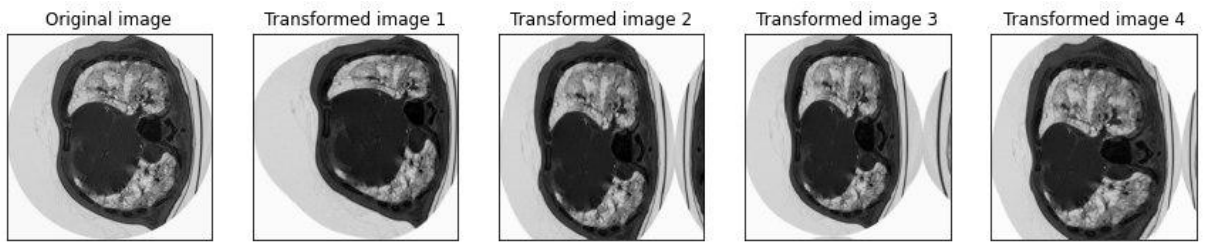


Рисунок 3.3 — Пример применения аугментации $G^1[0.3]$ к одному изображению

Эта аугментация также хорошо подходит именно для задач медицинской сегментации.

	$G^1[0.3]$	$G^1[0.37]$
Epochs	63	47
Loss	9.0378	9.0527
Accuracy	0.9902	0.9901
mIoU	0.7857	0.8016

Таблица 3.3 — Сравнение результатов модели на данных, аугментированных G^1 с различными параметрами

Видим довольно интересный результат: $G^1[0.3]$ показывает лучший результат с точки зрения Accuracy, а $G^1[0.37]$ – с точки зрения IoU.

$G^1[0.15]$ и $G^1[0.23]$ показали несравнимо худшие результаты.

Приведем сравнение простейших аугментаций с наилучшими параметрами. За E обозначим «единичную», тождественную, аугментацию:

	E	$V^{0.5}$	$V^{0.5}S^1$	B^1	$G^1[0.3]$	$G^1[0.37]$
Epochs	41	59	76	55	63	47
Loss	9.028	9.0246	9.0352	9.0113	9.0378	9.0527
Accuracy	0.9891	0.9896	0.9900	0.9892	0.9902	0.9901
Ground Glass IoU	0.6709	0.6958	0.7208	0.6366	0.7045	0.6970
Consolidation IoU	0.5052	0.4556	0.5659	0.5645	0.4967	0.5707
Lungs Other IoU	0.9407	0.9435	0.9441	0.9406	0.9467	0.9343
Background IoU	0.9946	0.9948	0.9945	0.9948	0.9950	0.9949

Таблица 3.4 — Сравнение простейших аугментаций

Видим, что у каждой простейшей аугментации есть свои сильные и слабые стороны. Очень хорошо показала себя ShiftScaleRotate, получив хорошие Ассурасу и IoU на патологиях. $G[0.3]$ хорошо научилась сегментировать более простые классы (легкие и фон), а $G[0.37]$ – классы патологий.

Попробуем совместить достоинства различных простейших аугментаций, рассмотрев их композицию.

Композиции простейших аугментаций

Пусть мы имеем аугментации X^p и Y^q . Объединение аугментаций будем обозначать как X^pY^q – т. е. мы сначала применяем к изображению X (с вероятностью p), затем Y (с вероятностью q) (хотя в нашем случае порядок применения аугментаций не важен, они коммутативны). Выбор из аугментаций будем обозначать как $Z^t = (X^p|Y^{1-p})^t$ – с вероятностью t мы применяем одну из аугментаций X , Y , причем с вероятностью p мы применяем X , а с вероятностью $(1 - p)$ – Y .

Сравним результаты моделей, обученных на различных композициях простейших аугментаций. Мы будем использовать только аугментации с наилучшими параметрами, поэтому далее обозначим $S^p = S^p[15, 0.05, 0.05]$, $B^p = B^p[0.1, 0.1]$, $G^p = G^p[0.3]$, $G_1^p = G^p[0.37]$.

Комбинировать аугментации также оказалось непростой задачей. Нередко комбинация приводит к ухудшению результатов модели на несколько сотых или десятых по сравнению как с простейшими аугментациями, так и с базовой неаугментированной моделью. Вероятно, причина в том, что комбинация слишком сильно искажает данные и их распределение, что приводит к тому, что модель не видит в нем закономерностей. Примеры таких неудачных композиций:

- $V^{0.5}S^{0.25}B^{0.5}G^{0.25}$;
- $V^{0.5}B^{0.7}(S^{0.5}|G^{0.5})^{0.5}$;
- $V^{0.5}(B^{0.5}|S^{0.25}|G^{0.25})^1$;

- $V^{0.5}B^1G^1$;
- $V^{0.5}B^{0.35}(S^{0.5}|G^{0.5})^{0.2}$.

Сравним композиции, показавшие себя лучше:

- $C_1 = V^{0.5}B^1(S^{0.5}|G^{0.5})^{0.5}$;
- $C_2 = V^{0.5}B^{0.5}(S^{0.5}|G^{0.5})^{0.3}$;
- $C_3 = V^{0.5}B^{0.5}(S^{0.5}|G_1^{0.5})^{0.3}$.

Добавим также колонки E (тождественная аугментация) и M – с наилучшими показателями из сравнения простейших аугментаций (таблица 3.4).

	E	M	C_1	C_2	C_3
Epochs	41	47	56	105	71
Loss	9.028	9.0113	9.0492	<i>9.0134</i>	9.0273
Accuracy	0.9891	0.99021	0.9896	<i>0.99018</i>	0.9900
Ground Glass IoU	0.6709	0.7208	0.6955	0.7086	<i>0.7097</i>
Consolidation IoU	0.5052	0.5707	0.5114	0.5568	<i>0.5604</i>
Lungs Other IoU	0.9407	0.9467	0.9415	<i>0.9455</i>	0.9453
Background IoU	0.9946	0.9950	0.9943	0.9952	0.9947

Таблица 3.5 — Сравнение композиций аугментаций. **Жирным** выделен лучший результат по всем колонкам, *курсивом* – лучший результат по колонкам C_1 , C_2 , C_3 .

Видим, что превзойти колонку M практически не удалось. Тем не менее, композиции в целом показывают лучший результат, чем каждая отдельная простейшая аугментация, сохраняя сильные стороны входящих в них компонентов. Что интересно, композиции перенимают особенности у входящих в них составных частей; так, C_2 , в состав которой входит G , лучше показывает себя на простых классах, а C_3 , в состав которой входит G_1 , – на патологиях. Такие же особенности мы подметили относительно самих G и G_1 .

Таким образом, нам удалось значительно улучшить результаты модели при помощи аугментаций, как простейших, так и их композиций. Простейшие аугментации дают более специализированные результаты (значительно улучшают лишь одну из многих метрик), тогда как композиции сохраняют достоинства и особенности входящих в них компонентов, улучшая результаты по всем метрикам. Чтобы найти подходящие параметры аугментаций и композиций, требуется рассмотреть разные варианты; многие будут работать хуже неаугментированной модели из-за смещения распределения данных, однако другие позволяют значительно улучшить показатели.

3.2 Архитектуры

До этого момента мы использовали Unet в качестве архитектуры своей нейронной сети. Рассмотрим эту архитектуру подробнее, а также сравним ее с альтернативами – другими архитектурами нейронных сетей для сегментации изображений.

3.2.1 Обзор архитектур нейронных сетей для сегментации изображений

Unet [12]. Одна из первых архитектур сегментации, часто лежащая в основе других, более сложных. Основная ее особенность заключается в том, что после типичных для сверточных нейронных сетей сворачивающих слоев, понижающих размеры картинки, идут «разворачивающие» слои. Они повышают разрешение и точность ответа, а также позволяют модели обучаться на комбинациях свернутых и развернутых данных. Получается U-образная структура, давшая сети название.

Структура была разработана специально для биомедицинских целей, поэтому учитывает особенности машинного обучения в этой области, а именно недостаток изображений, подходящих для обучения, и необходимость правильно локализовать классы на изображении. Еще одним ее достоинством является быстрая работа и обучение модели.

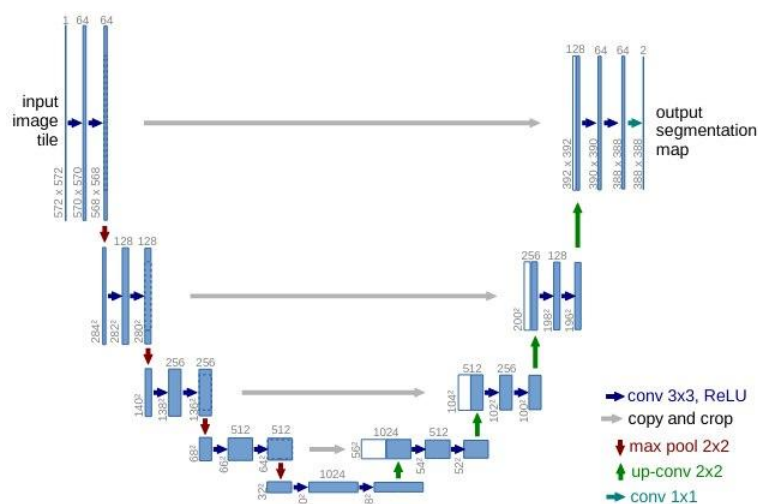


Рисунок 3.4 — Схема архитектуры Unet [12].

Unet++ [13]. Архитектура, в которой реализованы идеи, расширяющие и улучшающие Unet. Два основных отличия:

1. В Unet результаты сворачивающих слоев прямо объединяются с разворачивающими слоями. В Unet++ добавлены промежуточные сверточные слои (skip pathways), соединяющие «стенки» U-слоев (см.

рисунок 3.5). Согласно гипотезе создателей архитектуры, это позволяет семантически приблизить друг к другу нейроны сворачивающих и разворачивающих слоев, что улучшает работу оптимизатора.

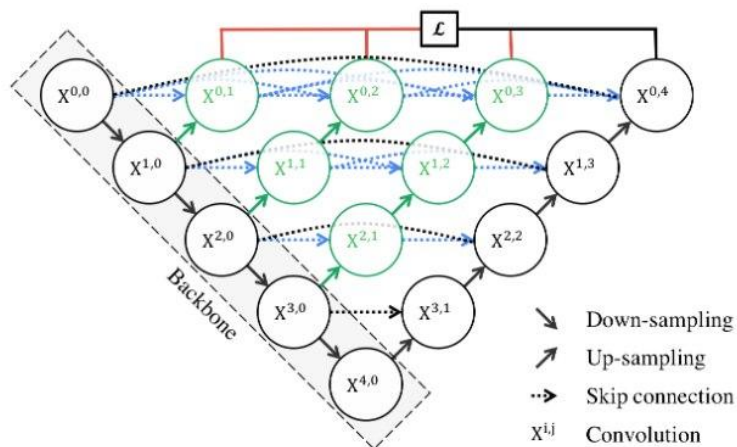


Рисунок 3.5 — Схема архитектуры Unet++ [13].
Черным обозначена Unet, зеленым и синим — «skip pathways».

- Используется так называемое Deep Supervision: к скрытым слоям добавляется скрытая функция потерь, которая позволяет увеличить влияние внутренних слоев на итоговое предсказание модели, что, в свою очередь, ускоряет ее сходимость.

MAnet (Multi-scale Attention Net) [14]. Также строится на основе Unet.

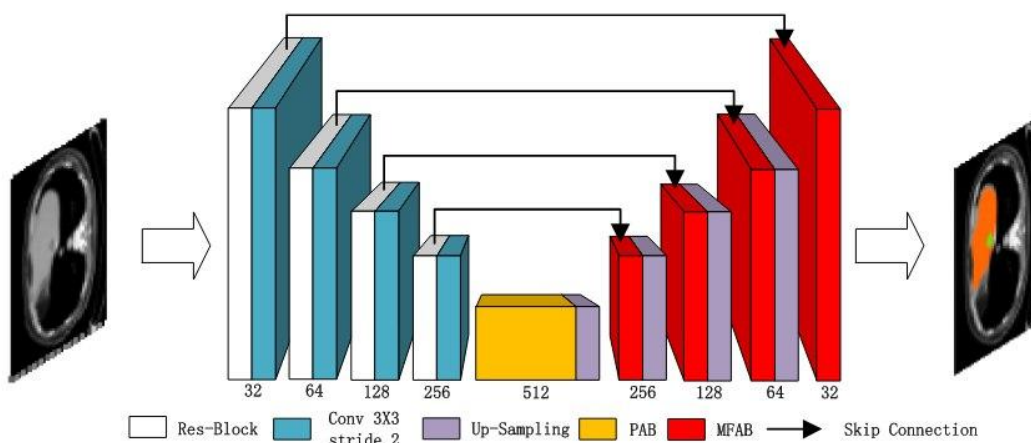


Рисунок 3.6 — Схема архитектуры MAnet [14].

Основные отличия:

- К сверточным слоям добавлены так называемые RES-блоки. Таким образом Unet комбинируется с остаточными нейронными сетями (ResNets [15]), целью которых является решение проблемы

затухающего градиента (так как изменения весов распределяются пропорционально между нейронами, большое количество нейронов приводит к ничтожно малым изменениям весов, и сеть быстро перестает учиться).

2. В переход между сверточными и разверточными слоями добавлен РА-блок (Position-wise Attention Block). Его предназначение – объединить информацию из многих локальных слоев в один, таким образом получив информацию из нескольких источников сразу. Согласно статье, сеть таким образом получает более глобальный взгляд на сегментируемое изображение, что приводит к лучшему пониманию различий между классами и большей согласованности слоев.
3. К разверточным слоям добавлены MFA-блоки (Multi-scale Fusion Attention Block). Их цель – настроить внимание модели аналогично человеческому восприятию, сместив фокус на определенную, нужную в данный момент информацию (значение признаков, нужных для сегментации, увеличивается, значение остальных признаков уменьшается). Таким образом увеличивается точность сегментации.

Deeplab3 [16]. Данная архитектура отличается от Unet и основана на других идеях, преследующих, однако, ту же цель, что и Unet: увеличить размер и качество изображения после (или во время) применения сверточных слоев. Для этого используются два метода:

1. «Atrous convolution» (*trous* – фр. отверстие, так что можно перевести как «дырявая свертка»). Свертка применяется не к соседним элементам матрицы (пикселям изображения), а к отстоящим друг от друга на величину Rate. Вместе с увеличением пробега (stride) такой метод позволяет сохранять исходные размеры изображения, имитируя при этом ситуации, схожие с теми, что возникают при применении сверточных слоев.

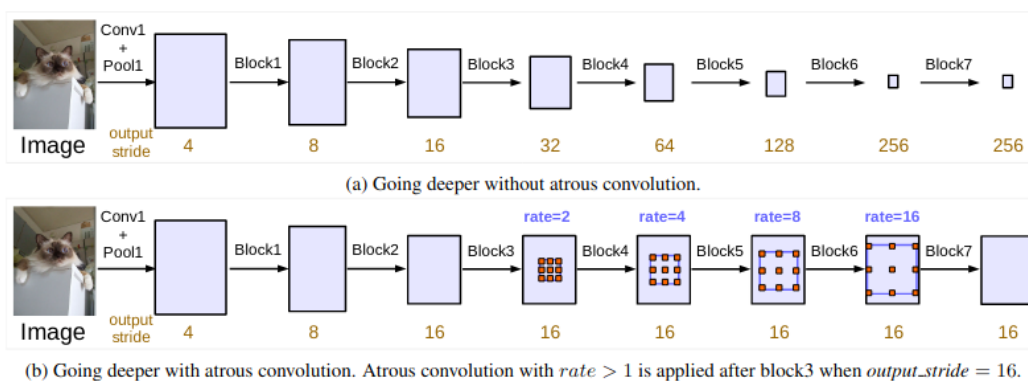


Рисунок 3.7 — Иллюстрация работы atrous convolution [16].

2. «Atrous Spacial Pyramid Pooling» (блок-пирамида). Несколько дырявых сверточных слоев разных размеров применяются к одной и той же матрице из предыдущего слоя. Таким образом сохраняются как сильные стороны анализа изображений малых размеров (улучшается качество классификации), так и сильные стороны анализа изображений больших размеров (сохраняется качество сегментации). Далее результаты работы данных слоев объединяются.

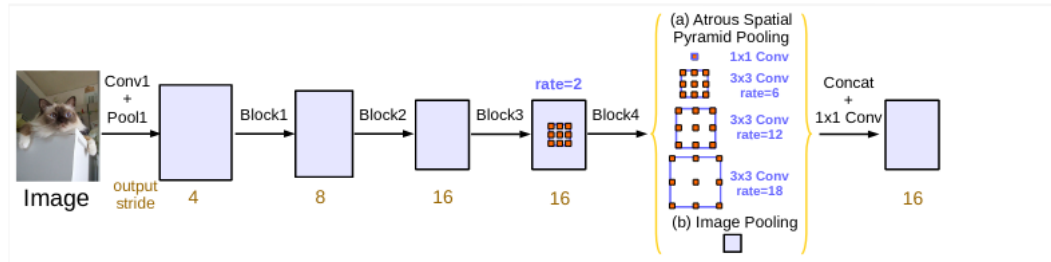


Рисунок 3.8 — Блок-пирамида [16].

Deeplab3+ [17]. Комбинирует идеи Deeplab3 и Unet: имеет U-образную структуру с пирамидными блоками.

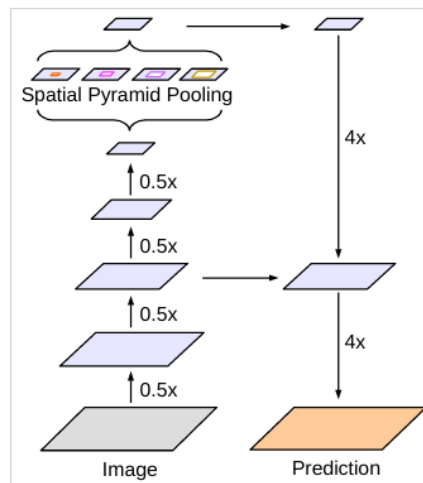


Рисунок 3.9 — Схема архитектуры Deeplab3+ [17].

3.2.2 Сравнение результатов архитектур в данной задаче

Тестирование проведем на неаугментированных моделях, так как параметры аугментаций мы подбирали под Unet, соответственно, для других архитектур аугментации с теми же параметрами могут быть не столь эффективны. Будем пользоваться реализациями архитектур из библиотеки `segmentation-models-pytorch`.

	Unet	Unet++	MAnet	Deeplab	Deeplab+
Epochs	41	27	38	56	53
Loss	9.0280	9.1174	9.0372	9.0482	9.0268
Accuracy	0.9891	0.9864	0.9890	0.9859	0.9881
Ground Glass IoU	0.6709	0.5578	0.6774	0.6160	0.6609
Consolidation IoU	0.5052	0.0624	0.4523	0.4962	0.5204
Lungs Other IoU	0.9407	0.9233	0.9400	0.9215	0.9381
Background IoU	0.9946	0.9944	0.9947	0.9919	0.9942

Таблица 3.6 — Сравнение архитектур сегментации

Можно сделать несколько достаточно интересных выводов:

- Несмотря на то, что Unet является самой простой сетью (большинство других архитектур добавляют слои и блоки к существующей U-структуре), ее результаты оказались одними из лучших. Это крайне нетипичный результат, говорящий, возможно, о некоторой специфичности данной задачи и качестве набора данных. Можно также сделать вывод о том, что в машинном обучении часто требуется пробовать неочевидные средства (выбирать более простую архитектуру вместо более сложной).
- Хуже всего оказались Unet++ и Deeplab. Также отметим, что обучение данных сетей длится дольше всего и требует наибольшего количества вычислительных ресурсов в сравнении с другими архитектурами. Можно сделать вывод, что наша задача требует более простого строения архитектуры, что позволяет ей быстро выделять требуемую информацию. Кроме того, видим, что Unet-основа положительно влияет на результаты сетей, что особенно хорошо видно при сравнении Deeplab и Deeplab+.
- Говоря о выделении требуемой информации: именно эта идея является центральной в строении MAnet, которая показывает хороший результат в аугментации отдельных классов. Вероятно, если обучать ее на аугментациях, можно получить как минимум сравнимый с Unet результат, а как максимум – и лучший. Также видим, что на обучение MAnet потребовалось достаточно мало эпох, что можно связать с наличием RES-блоков.
- Довольно интересный результат видим у Deeplab+: сеть максимизировала результаты на классе, которого в выборке меньше всего, и получила на нем оценку, значительно превосходящую оценки других архитектур. Можно предположить, что пирамидные блоки особенно хорошо позволяют классифицировать классы патологий (что, вообще говоря, является главной задачей в медицинской сегментации).

4. Заключение

Задача медицинской сегментации при помощи машинного обучения на данный момент занимает многих исследователей. У нее есть большой потенциал и возможность значительно улучшить качество здравоохранения. Способы ее решения активно развиваются; так, архитектура Deeplab+ была опубликована в 2018 году, а MAnet – и вовсе в 2020. Наверняка развитие этой области не остановится, и вскоре мы увидим новые методы и идеи.

Мы исследовали методы сегментации изображений на снимках легких; задачей было найти на них патологии, вызываемые COVID-19.

Был рассмотрен способ улучшить результаты модели при помощи аугментаций. Мы исследовали различия простейших аугментаций и их композиций. Можно сказать, что и те, и другие позволяют значительно улучшить качество сегментации, при этом каждая имеет свои особенности; композиции аугментаций сохраняют сильные и слабые стороны своих компонентов.

Мы рассмотрели ведущие архитектуры нейронных сетей медицинской сегментации и сравнили их работу на текущей задаче. Полученные результаты достаточно интересны и специфичны для нашего исследования; так, более сложная Unet++ показала себя значительно хуже Unet, на основе которой она построена. Мы получили, что у каждой архитектуры имеются свои особенности, что достигается за счет специальных блоков в их строении; так, MFA-блоки в MAnet позволяют лучше сегментировать отдельные специфичные классы, а пирамидные блоки в Deeplab3+ позволяют подмечать особенности, недоступные другим архитектурам.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Hepatoid adenocarcinoma of the stomach is a special and easily misdiagnosed or missed diagnosed subtype of gastric cancer with poor prognosis but curative for patients of pN0/1: the experience of a single center / Yibin Xie, Zhonghai Zhao, Peng Li, Yang Wang, Chunguang Guo, Xiaoliang Wang, Wei Tang, Qian Liu, Ning Lu, Liyan Xue, Dongbing Zhao // Int J Clin Exp – 2015. Т. 8, №5 – С. 6762–6772.
2. COVID-19 CT segmentation dataset [Электрон. ресурс]. – <http://medicalsegmentation.com/covid19/>
3. COVID-19 CT Images Segmentation | Kaggle [Электрон. ресурс]. – <https://www.kaggle.com/c/covid-segmentation/overview>
4. Amini B., Bell D. Ground-glass opacification [Электрон. ресурс]. // Reference article, Radiopaedia.org. – 2008. – <https://radiopaedia.org/articles/1404>
5. Babu, V. Lung consolidation [Электрон. ресурс]. // Case study, Radiopaedia.org. – 2018. – <https://radiopaedia.org/cases/58923>
6. COVID-19 DATABASE [Электрон. ресурс]. // Società Italiana di Radiologia Medica e Interventistica. – <https://sirm.org/category/senza-categoria/covid-19/>
7. Bell, D. COVID-19. [Электрон. ресурс]. // Reference article, Radiopaedia.org. – 2020. – <https://radiopaedia.org/articles/73913>
8. PyTorch [Электрон. ресурс]. – <https://pytorch.org/>
9. Segmentation Models with pretrained backbones. Pytorch [Электрон. ресурс]. – https://github.com/qubvel/segmentation_models_pytorch
10. Data Augmentation Using Learned Transformations for One-Shot Medical Image Segmentation / Amy Zhao, Guha Balakrishnan, Fredo Durand, John V. Guttag, Adrian V. Dalca // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) – 2019. – С. 8543–8553.
11. Albumentations: fast and flexible image augmentations [Электрон. ресурс]. – <https://albumentations.ai/>
12. U-Net: Convolutional Networks for Biomedical Image Segmentation / Olaf Ronneberger, Philipp Fischer, Thomas Brox // arXiv – 2015. №1505.04597 – 8 с.

13. UNet++: A Nested U-Net Architecture for Medical Image Segmentation / Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, Jianming Liang // arXiv – 2018. №1807.10165 – 8 c.
14. MA-Net: A Multi-Scale Attention Network for Liver and Tumor Segmentation / Tongle Fan, Guanglei Wang, Yan Li, Hongrui Wang // IEEE Access – 2020. T. 8 – C. 179656-179665.
15. Deep Residual Learning for Image Recognition / Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun // IEEE Conference on Computer Vision and Pattern Recognition (CVPR) – 2016. – C. 770-778.
16. Rethinking Atrous Convolution for Semantic Image Segmentation / Liang-Chieh Chen, George Papandreou, Florian Schroff, Hartwig Adam // arXiv – 2017. №1706.05587 – 14 c.
17. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation / Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, Hartwig Adam // arXiv – 2018. №1802.02611 – 18 c.