

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**  
**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**  
**ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И**  
**ИНФОРМАТИКИ**

**Кафедра биомедицинской информатики**

**ДЕТЕКТИРОВАНИЕ УЧАСТКОВ ЗАДАННОГО ТИПА НА**  
**РЕНТГЕНОВСКИХ ИЗОБРАЖЕНИЯХ ГРУДНОЙ КЛЕТКИ**

Курсовой проект

Синявского Тимура Владимировича  
студента 3 курса  
специальность «информатика»

Научный руководитель:  
доцент кафедры БМИ  
Ковалёв В.А.

Минск

2019

# ОГЛАВЛЕНИЕ

Глава 1. Постановка задачи.....	3
1.1 Генеративно-состязательные сети.....	3
1.2 Формулировка задачи.....	4
Глава 2. Решение задачи с помощью сегментации.....	5
2.1 Задача сегментации.....	5
2.2 Применение сегментации к поставленной задаче.....	7
2.3 Формирование признаков для отсеивания негодных снимков.....	7
2.4 Дерево решений и логистическая регрессия.....	8
2.5 Построение модели для отсеивания негодных снимков.....	9
2.6 Итоговый пайплайн отбора снимков.....	10
Глава 3. Другой подход к решению задачи.....	11
3.1 Transfer learning.....	11
3.2 Grad-CAM.....	12
3.3 Применение transfer learning и Grad-CAM для детектирования.....	13
Заключение.....	15
Список использованных источников.....	16

# Глава 1. Постановка задачи.

## 1.1 Генеративно-сопоставительные сети.

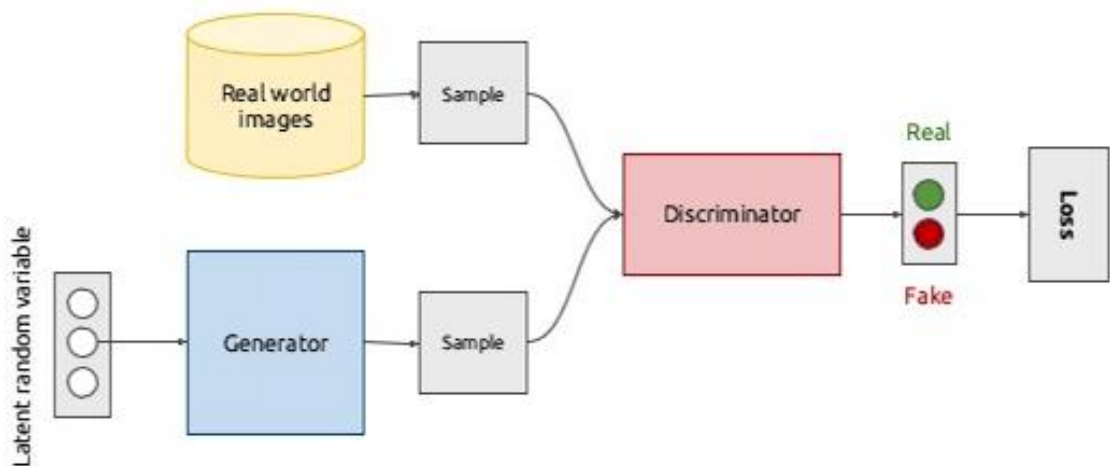
Генеративно-сопоставительная сеть (Generative Adversarial Network, GAN) [1] – это алгоритм глубокого обучения без учителя. GAN применяется для генерации изображений, музыки, речи, видео, трёхмерных моделей и др.

### Структура GAN.

GAN состоит из двух взаимодействующих нейронных сетей:

Генератор (generator) – принимает на вход случайный элемент латентного пространства и генерирует образец. Цель генератора – научиться создавать образцы, похожие на образцы из обучающей выборки (но не повторяющие их) и достаточно реалистичные для того, чтобы «обмануть» сеть-дискриминатор. Таким образом, задача генератора – не копировать уже существующие образцы, а создавать изображения с распределением настолько близким к распределению исходных данных, что дискриминатор не сможет их различить.

Дискриминатор (discriminator) – принимает на вход образец и оценивает реалистичность относительно не сгенерированных образцов. Цель дискриминатора – научиться отличать синтезированные генератором образцы от настоящих образцов из обучающей выборки. Другими словами, дискриминатор решает задачу бинарной классификации для поступающего ему на вход образца: определяет, является ли он «подлинным» или сгенерированным. При этом дискриминатор адаптируется к тому, что генератор создаёт всё более реалистичные образцы.



*GAN architecture.*

## Обучение GAN.

Принцип работы GAN иногда описывается с помощью метафор: первая сеть представляется как фальсификатор, а вторая – как эксперт, который сравнивает подделки с подлинником и указывает фальсификатору, какие детали в его работе реалистичны, а какие нет. Со временем фальсификатор учится создавать всё более реалистичные подделки, а эксперт становится всё более опытным в различении подделок. Генератор и дискриминатор соревнуются друг с другом, то есть между ними возникает антагонистическая игра.

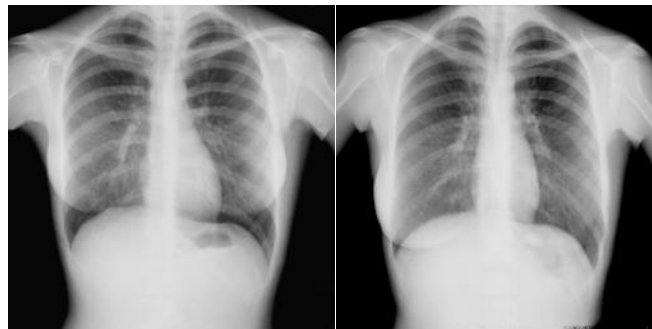
Генератор получает случайные вектора и генерирует образцы. Далее эти образцы наряду с реальными образцами из обучающей выборки подаются дискриминатору. А он возвращает вероятности принадлежности образцов к классам “сгенерированных” и “реальных”. Исходя из ошибок дискриминатора генератор модифицируется так, чтобы дискриминатор не смог отличить реальные образцы от сгенерированных. А после дискриминатор модифицируется так, чтобы снова отличать сгенерированное. То есть генератор и дискриминатор обучаются отдельно, но в рамках одной сети. Таким образом целью генератора является повысить процент ошибок дискриминатора, а целью дискриминатора является наоборот улучшение точности распознавания. Постепенно искусственные образцы на выходе генератора становятся всё более качественными.

На сегодняшний день GANы сильно развиваются, появляется много модификаций [2]. Одно из последних многочисленных достижений: StyleGAN [3] генерирует фотографии несуществующих людей в высоком разрешении [4].

### 1.2 Формулировка задачи.

Дана модель GAN, которая генерирует рентгеновские снимки грудной клетки мужчин и женщин размера 256 на 256 пикселей. Но женские снимки иногда генерируются несимметричные в области груди. Цель курсового проекта – научиться отсеивать такие снимки.

Пример хорошей (слева) и плохой (справа) генерации:



## **Глава 2. Решение задачи с помощью сегментации.**

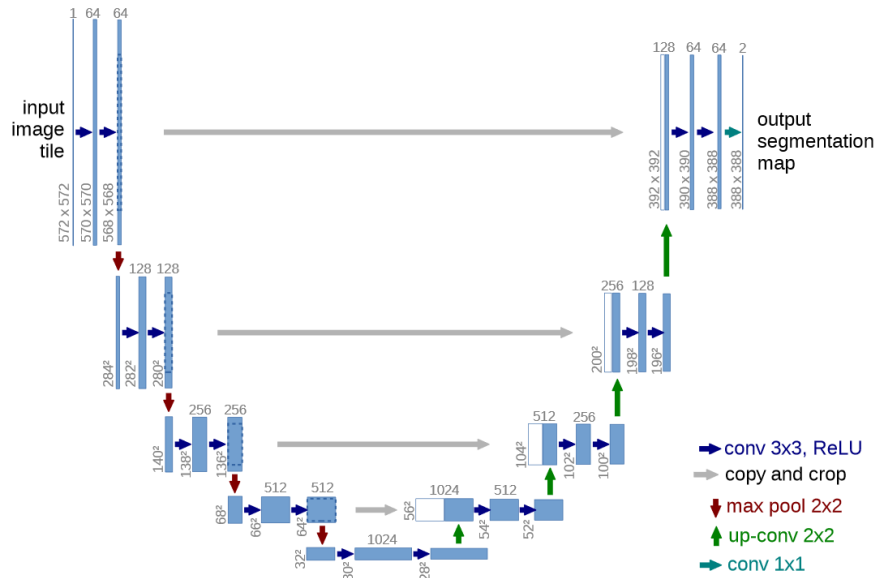
### **2.1 Задача сегментации.**

Сегментация - это процесс разделения изображения на несколько сегментов (множество пикселей, также называемых суперпикселями). Альтернативное определение сегментации: процесс присвоения меток каждому пикселю изображения таким образом, что пиксели с одинаковыми метками имеют общие визуальные характеристики. Цель сегментации заключается в упрощении изображения, чтобы его было легче анализировать. Сегментация изображений обычно используется для того, чтобы выделить объекты и границы на изображениях. Результатом сегментации изображения является множество сегментов, которые вместе покрывают всё изображение, или множество контуров, выделенных из изображения.

Существуют следующие алгоритмы сегментации [5]:

- Сегментация по регионам (пороговая сегментация, сегментация регионального роста) - разделяет объекты на разные регионы на основе некоторых пороговых значений.
- Сегментация с обнаружением краев (операторы Собеля и Лапласа) - использует прерывистые локальные элементы изображения для обнаружения краев и, следовательно, определения границы объекта.
- Сегментация на основе кластеризации - делит пиксели изображения на однородные кластеры.
- Сегментация с использованием нейронных сетей.

U-net [6] – это одна из нейросетевых архитектур используемых для сегментации. Впервые данная архитектура была применена к сегментации медицинских изображений, но позже так же распространилась и на другие виды изображений.



*U-net architecture.*

U-net состоит из двух частей. Первая часть – кодирующая часть (encoder). Она используется для захвата контекста изображения. Обычно кодировщик состоит из последовательности свёрточных слоёв и пулингов. Вторая часть сети – декодирующая часть (decoder). Декодер чаще всего симметричен кодировщику и обеспечивает точную локализацию на выходе. Таким образом U-net – это полностью свёрточная сеть (Fully Convolutional Network, FCN), то есть она содержит только свёрточные слои и не содержит полносвязных слоёв. Если задача – выделение одного объекта, то есть сегментация на два сегмента: “искомый объект” и “другое”, то на выходе модель каждому пикселю сопоставляет вероятность принадлежности к сегменту “искомый объект”. Так как задачу можно переформулировать в задачу классификации каждого пикселя, при обучении в качестве функции ошибки чаще всего используется бинарная кросс энтропия (Binary Cross Entropy, BCE)

$$BCE = -\frac{1}{N} \sum_{i=1}^N y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(1 - p(y_i))$$

Метрики для оценки модели сегментации:

- Попиксельная точность – доля верно предсказанных пикселей.
- Intersection over union (IoU, коэффициент Жаккара):

$$IoU = \frac{S_{pred \cap truth}}{S_{pred \cup truth}}$$

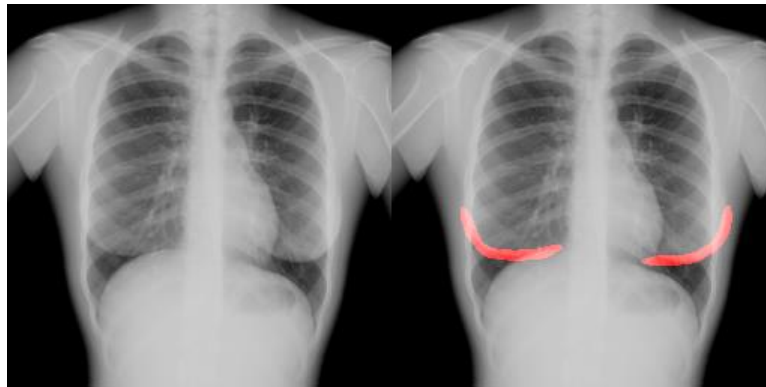
- Dice coefficient (F1 score, коэффициент Сёренсена):

$$DC = \frac{2 * S_{pred \cap truth}}{S_{pred} + S_{truth}}$$

## 2.2 Применение сегментации к поставленной задаче.

Для дальнейшего отсеивания снимков была решена задача сегментации женской груди на рентген снимке. Для решения использовалась модель U-net. Для обучения U-net было размечено порядка трехсот сгенерированных женских снимков. На вход модель принимает снимок размерности  $1 \times 256 \times 256$  (1 – размерность канала свёртки, так как снимок подаётся в чёрно-белом формате), значения пикселей переведены в диапазон от нуля до единицы. На выходе получается тензор той же размерности. Каждое значение тензора - вероятность принадлежности соответствующего пикселя к маске. Итоговая маска (состоящая из нулей и единиц) получается бинаризацией по порогу равному 0.5.

Пример результата работы модели (слева вход, справа полученная маска наложенная на исходный снимок):



## 2.3 Формирование признаков для отсеивания негодных снимков.

Признаки формируются на основе масок сегментации следующим образом:

1. Каждая маска разбивается на две равные части относительно центральной вертикальной оси снимка.
2. Правая половина отражается относительно вертикальной оси.
3. К каждой части применяется морфологическая операция *dilate* (операция увеличивает светлые области).
4. Считаются следующие коэффициенты (признаки) как для исходных половин, так и для половин после операции *dilate*:

$$IoU = \frac{S_{left \cap right}}{S_{left \cup right}}$$
$$\cos = \frac{left \cdot right}{\|left\| \|right\|}$$

$$area\ ratio = (\frac{S_{left}}{S_{left} + S_{right}} - \frac{1}{2})^2$$

Таким образом для каждого снимка получим шесть признаков. Интуитивный смысл первых двух признаков – мера похожести левой и правой части маски, а значит и мера симметрии исходного снимка. Третий признак – наоборот.

## 2.4 Дерево решений и логистическая регрессия.

Дерево решений классифицирует данное наблюдение при помощи последовательности вопросов, в которой следующий заданный вопрос зависит от ответа на текущий. Структура дерева представляет собой «листья» и «ветки». На рёбрах («ветках») дерева решения записаны атрибуты, от которых зависит целевая функция, в «листьях» записаны значения целевой функции, а в остальных узлах - атрибуты, по которым различаются случаи. Чтобы классифицировать новый случай, надо спуститься по дереву до листа и выдать соответствующее значение.

В основе популярных алгоритмов построения дерева решений, лежит принцип жадной максимизации прироста информации - на каждом шаге выбирается тот признак, при разделении по которому прирост информации оказывается наибольшим. Далее процедура повторяется рекурсивно, пока энтропия не окажется равной нулю или какой-то малой величине.

Плюсы дерева решений:

- Интерпретируемость.
- Быстрый процесс обучения и прогнозирования.
- Малое число гиперпараметров.

Минусы:

- Чувствительны к шумам во входных данных.
- Нестабильность - небольшие изменения в данных могут существенно изменять построенное дерево решений.

Логистическая регрессия – это статистическая модель, используемая для предсказания вероятности возникновения некоторого события путём подгонки данных к логистической кривой. Основная идея заключается в том, что признаковое пространство может быть разделено гиперплоскостью на два полупространства, в каждом из которых прогнозируется одно из двух значений целевого класса. Логистическая регрессия выдает вероятности отнесения к

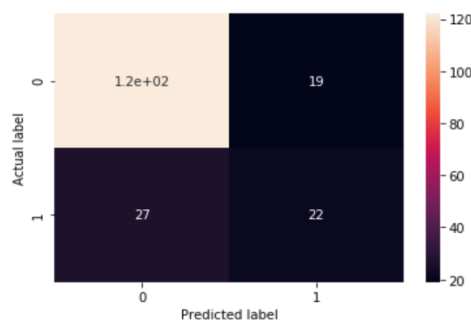
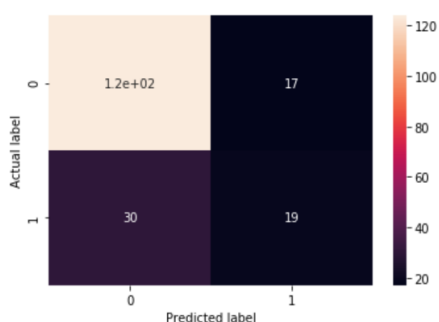


разным классам. Плюсы логистической регрессии такие же, как и у дерева решений. Минус: плохо работает в задачах, в которых зависимость ответов от признаков нелинейная.

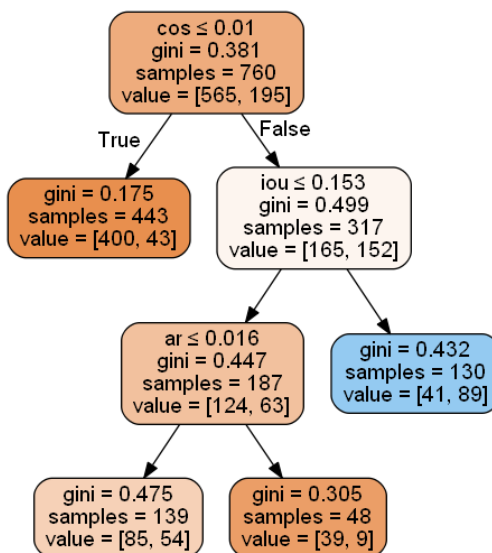
## 2.5 Построение модели для отсеивания негодных снимков.

Отсеивание плохих (несимметричных) снимков рассмотрим как задачу классификации: “хороший” или “плохой” снимок. Для решения были рассмотрены логистическая регрессия и дерево решений. На вход модели подаются признаки из п. 2.3. Для обучения этих алгоритмов было размечено 1000 снимков. Параметры моделей подбирались на кросс-валидации. Результаты:

DecisionTreeClassifier - score\_train: 0.817, score\_test: 0.753 LogisticRegression - score\_train: 0.807, score\_test: 0.758  
Test confusion matrix: Test confusion matrix:



Визуализация обученного дерева решений (оранжевые узлы – плохие снимки, синие – хорошие):



Видно, что задача классификации решается не очень хорошо. Отсюда следует, что признаков, полученных из масок сегментации, недостаточно. Но тем не менее исходное соотношение плохих снимков к хорошим почти 3 к 1, а после отсеивания - 1 к 1.

## **2.6 Итоговый пайплайн отбора снимков.**

1. Снимки прогоняются через модель сегментации U-net и получаются бинарные маски.
2. Каждой маске в соответствие ставятся признаки симметрии.
3. К признакам применяется классифицирующая модель, которая принимает решение о пригодности снимка.

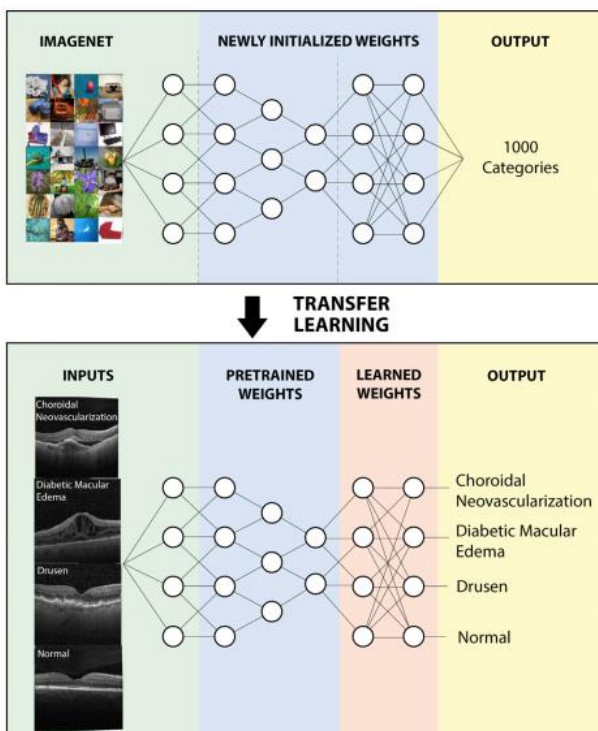
## Глава 3. Другой подход к решению задачи.

### 3.1 Transfer learning.

Transfer learning – это метод машинного обучения, в котором модель, разработанная для определенной задачи, используется как база для построения модели для другой задачи. В глубоком обучении предварительно обученные нейронные сети часто используются в компьютерном зрении и обработке языка. Это помогает сэкономить большие вычислительные и временные ресурсы для разработки моделей для этих задач с нуля. Так же это позволяет быстрее улучшать качество модели на поставленной задаче путем передачи знаний из связанной задачи, которая уже изучена. Но этот метод работает только если признаки, изученные в базовой задаче, являются достаточно обобщёнными, то есть подходящими как для базовой, так и для целевой задачи, а не специфичны для базовой задачи.

В компьютерном зрении одна из наиболее распространенных базовых задач - это классификация изображений. ImageNet - набор данных изображений с метками от 1 до 1000 (каждая метка – класс, например человек, машина и т.п.). Нейросетевые модели, обученные на этом датасете, часто используются для transfer learning. Примеры таких моделей: ResNet [7], VGG, Inception. Они

обучаются днями/неделями на мощном оборудовании, что не всегда доступно обычному разработчику. Эти модели могут быть использованы как базовые для моделей, которые так же принимают на вход изображения. Признаки на ранних слоях сети более общие (например, контуры, формы и т.п.), а на поздних – более специфичные для базовой задачи. Поэтому зачастую при дообучении на новую задачу ранние слои базовой сети замораживаются (т.е. веса в этих слоях не обновляются при обучении), а поздние обновляются или вовсе инициализируются случайными значениями. Этот процесс называется fine-tuning.



В обработке языка используются эмбединги, которые отображают слова в многомерное векторное пространство. Примеры таких моделей: word2vec и GloVe.

Таким образом transfer learning даёт следующие преимущества:

- Результат получается лучше.
- Существенно сокращается время, которое необходимо потратить на обучение модели.
- Позволяет использовать меньше данных для целевой задачи.

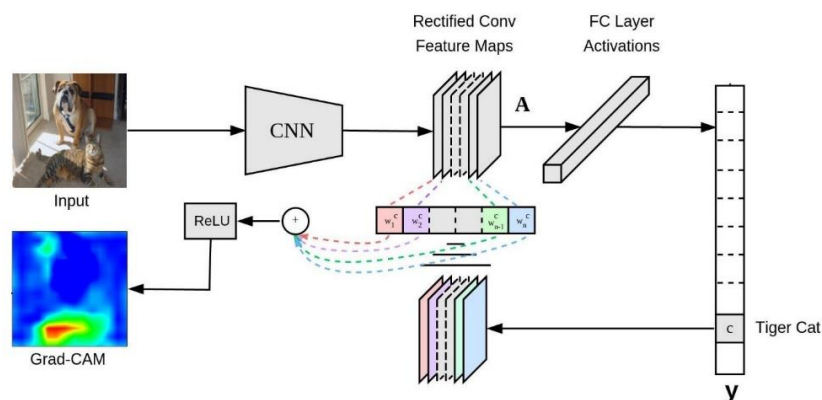
Но это выполняется при условии, что признаки, выученные на базовой задаче, применимы к поставленной задаче.

### 3.2 Grad-CAM.

Gradient-weighted Class Activation Mapping (Grad-CAM) [8] – алгоритм интерпретации решений свёрточных нейронных сетей за счёт выделения областей входного изображения, которые “важны” при формировании моделью ответа. Выделение получается в виде тепло карт (heat-map, localization map).

Результат получается следующим образом. Сначала считаются производные выходного нейрона требуемого класса  $y^c$  ( $c$  – индекс класса) по значениям feature map  $A_{ij}^k$  (обычно это выход последнего свёрточного слоя, имеем  $K$  карт, поэтому  $k = \overline{1, K}$ ). Получаем  $K$  матриц производных такой же размерности как и  $A^k$ . Для каждой такой матрицы считаем среднее значение её элементов и получаем веса  $\alpha_k^c$ . Далее берем линейную комбинацию  $A^k$  с весами  $\alpha_k^c$  и применяем функцию  $ReLU(x) = \max(0, x)$ . Получаем одну матрицу  $L^c$ , которую растягиваем до размеров входного изображения с помощью интерполяции или других методов и накладываем на входное изображение.

$$\alpha_k^c = \overbrace{\frac{1}{Z} \sum_i \sum_j}^{\text{global average pooling}} \underbrace{\frac{\partial y^c}{\partial A_{ij}^k}}_{\text{gradients via backprop}} \quad L_{\text{Grad-CAM}}^c = ReLU \left( \underbrace{\sum_k \alpha_k^c A^k}_{\text{linear combination}} \right)$$



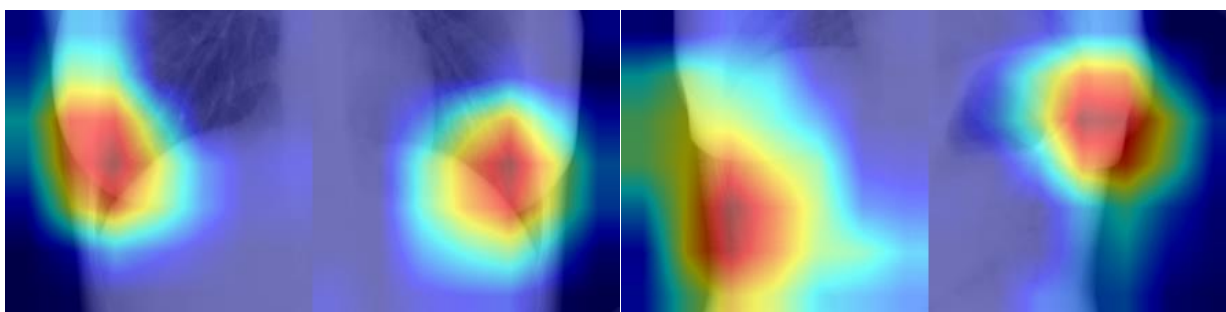
Работу нейронных сетей часто ассоциируют с черным ящиком. Данный метод является очень важным для понимания решения алгоритма, так как позволяет проинтерпретировать решение широкого класса свёрточных нейронных сетей (CNN).

### 3.3 Применение transfer learning и Grad-CAM для детектирования.

Идея заключается в следующем: обучим нейросетевой классификатор для определения пола человека по рентген снимку. А потом посмотрим на что сеть обращает внимание при определении пола методом Grad-CAM.

В качестве классификатора использовалась предобученная на ImageNet модель ResNet-18. Все слои кроме последних были заморожены. А последние полносвязные слои были заменены на один слой с одним выходным нейроном и логистической функцией активации. Сеть выдаёт как ответ единицу, если на снимке женщина, и ноль, если мужчина. При обучении использовалось порядка 200 снимков женщин и столько же мужских снимков. Использовался следующий препроцессинг: снимки обрезались сверху, так как хотелось бы, чтоб сеть делала решение на основе грудной области, а не шеи или плеч. Так же на вход сети подавались левые и правые половины снимков по отдельности, чтобы избежать ответа сети только по одной половине в случае подачи целого снимка. После обучения сеть показала точность 90%.

Применение Grad-CAM к обученной сети дало следующие результаты:



К сожалению, полученные хитмапы получаются довольно зашумлёнными. Например, так как в обучающей выборке на женских снимках иногда фигурировала рука, опёртая на бок, а среди мужских снимков такого не наблюдается, при получении хитмапа алгоритм Grad-CAM с большой вероятностью подсветит руку. Таким образом извлечение информации о положении груди или симметрии рентген снимка затрудняется.

## **ЗАКЛЮЧЕНИЕ**

Таким образом в ходе выполнения курсового проекта были рассмотрены различные подходы к задаче детектирования на рентгенографических изображениях. И был разработан пайплайн на основе детектирования с помощью сегментации, который улучшает генерацию предоставленной генеративно состязательной сети путём отсеивания некачественных изображений по признакам асимметрии детектированных областей.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- [1] “Generative Adversarial Nets” [Electronic resource]. - <https://arxiv.org/pdf/1406.2661.pdf>
- [2] “The GAN Zoo” [Electronic resource]. - <https://github.com/hindupuravinash/the-gan-zoo>
- [3] “A Style-Based Generator Architecture for Generative Adversarial Networks” [Electronic resource]. - <https://arxiv.org/pdf/1812.04948.pdf>
- [4] [Electronic resource]. - <https://thispersondoesnotexist.com/>
- [5] “Image Segmentation Algorithms Overview” [Electronic resource]. - <https://arxiv.org/ftp/arxiv/papers/1707/1707.02051.pdf>
- [6] “U-Net: Convolutional Networks for Biomedical Image Segmentation” [Electronic resource]. - <https://arxiv.org/pdf/1505.04597.pdf>
- [7] “Deep Residual Learning for Image Recognition” [Electronic resource]. - <https://arxiv.org/pdf/1512.03385.pdf>
- [8] “Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization” [Electronic resource]. - <https://arxiv.org/pdf/1610.02391.pdf>