

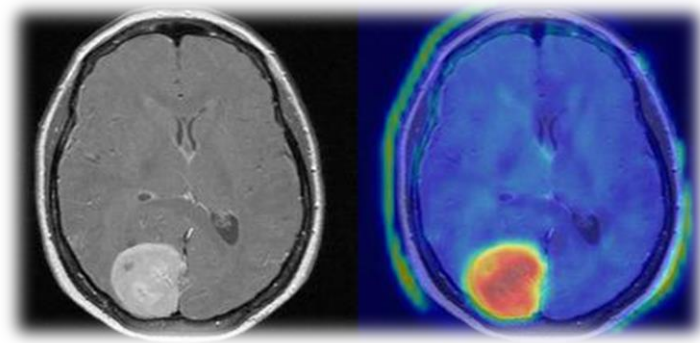
Многофункциональный модуль поиска заболеваний



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ

Яндекс

- Индивидуальный программный проект
- Выполнил студент: Ромашкина Арина Дмитриевна, БПМИ 223
- Научный руководитель: Боровский Андрей Олегович, Научный сотрудник Факультета Компьютерных Наук



Постановка задачи

ЦЕЛЬ: создать телеграмм-бота, возвращающего предполагаемое заключение, а также картинки визуализации работы модели

Создание модели с помощью pytorch для задачи классификации

Изучить и применить фреймворк Optuna для построения модели

Изучить и применить дистилляцию знаний

Изучить и реализовать метод интерпретации LRP

Изучить метод интерпретации Grad-CAM

Создание телеграмм-бота

Телеграмм-бот



@DiseaseIdentificationModuleBot

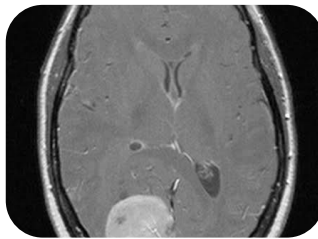


Папка с картинками
для тестирования бота

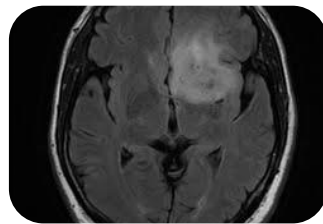
<https://disk.yandex.ru/d/vTghYEzRhZRliQ>

Датасет Опухолей

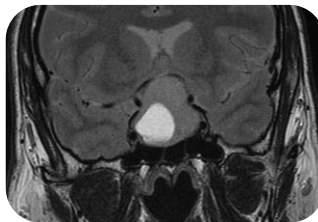
- BRAIN TUMORS с сайта Kaggle – классификация опухолей
- 2870 тренировочных и 394 тестовых изображений
- 4 класса:
 - менингиома
 - глиома
 - опухоль гипофиза
 - нет опухоли



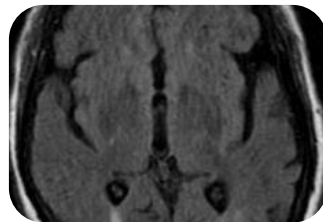
менингиома



глиома



опухоль
гипофиза

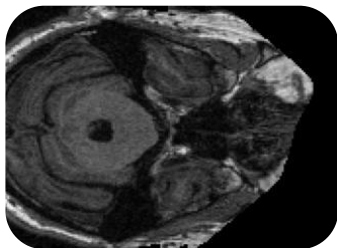


нет опухоли

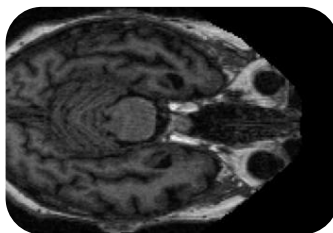
Датасет Деменции

- OASIS с сайта Kaggle – классификация стадии деменции
- 85987 изображений, 347 пациентов
- 4 класса:
 - Очень легкая стадия
 - Легкая стадия
 - Умеренная стадия
 - Нет деменции

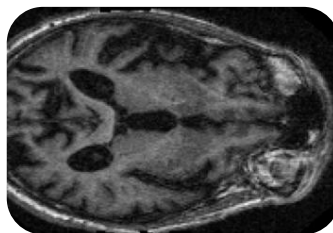
Число Пациентов	Обучающая	Тестовая
Нет деменции	244	22
Очень легкая	46	12
Легкая	17	4
Умеренная	1	1



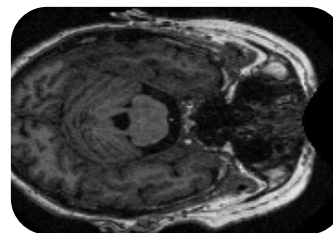
Очень легкая
стадия



Легкая
стадия



Умеренная
стадия

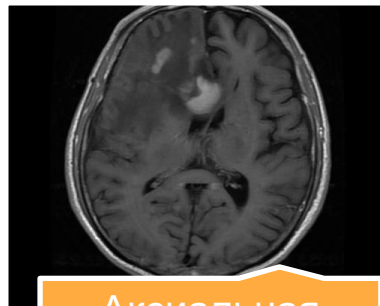


Нет
деменции

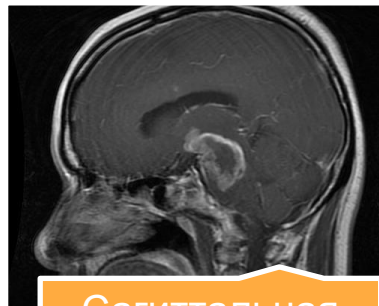
Аугментация данных

Основные особенности данных:

- Снимки МРТ проводят с некоторой периодичностью
- Разные снимки пациента не могут попасть одновременно в тестовую и тренировочную выборки
- снимки сделаны в разных проекциях, представлены разные срезы головного мозга
- Локализация опухоли/изменений определяет класс, и зависит относительно центра/расположения в нужной доле – нельзя использовать Flip, Crop и тд



Аксиальная проекция

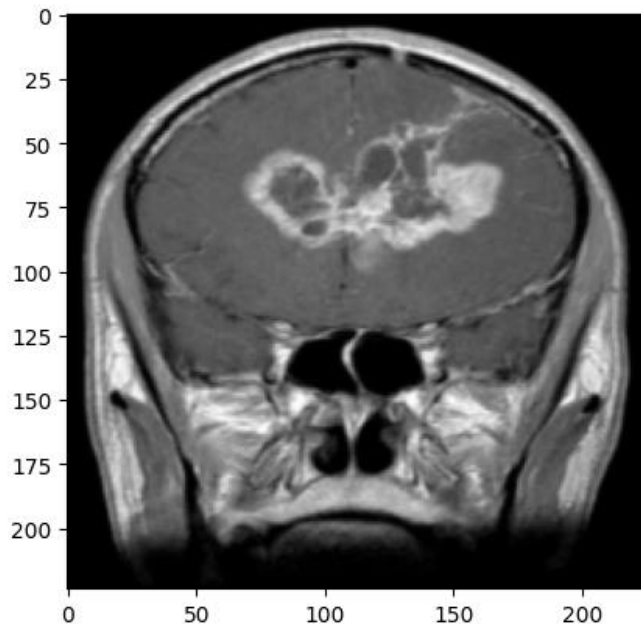


Сагиттальная проекция



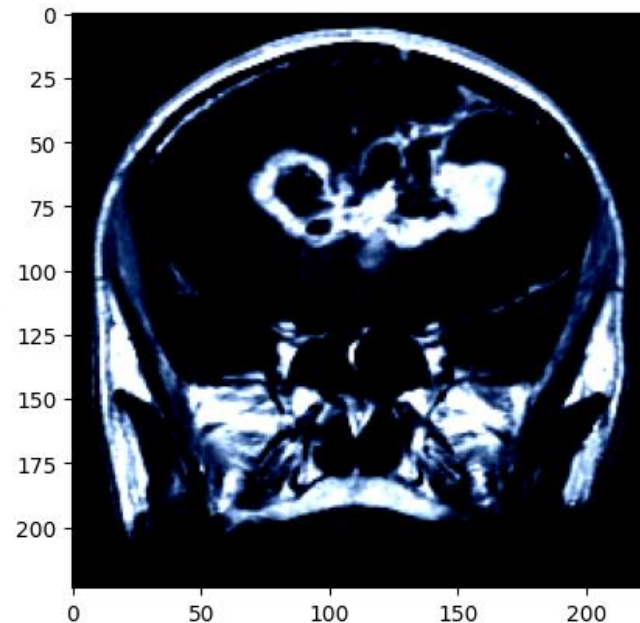
Коронарная проекция

Аугментация данных

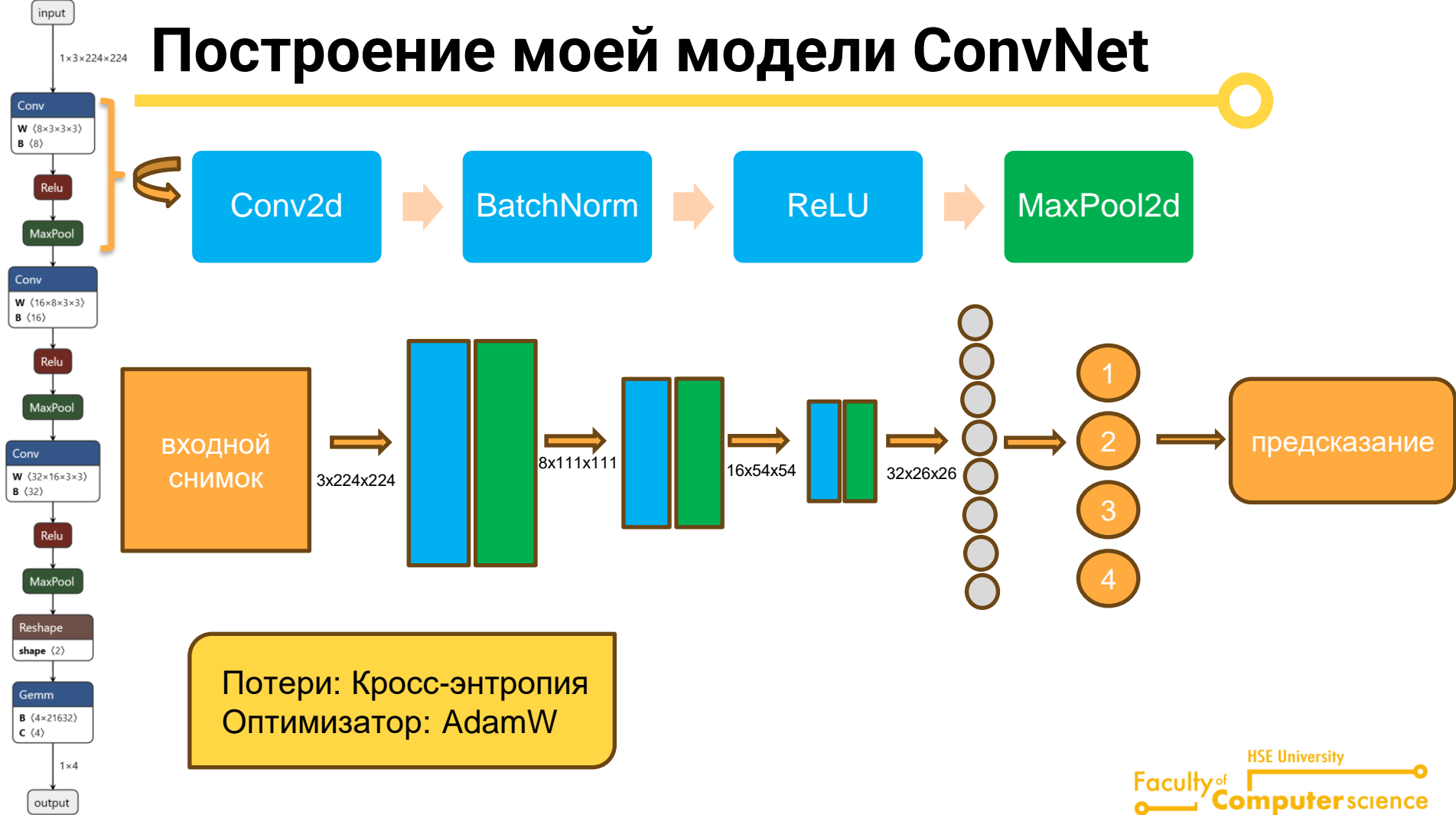


аугментация

Resize(224, 224)
Normalize(
mean=[0.485,
0.456, 0.406],
std=[0.229, 0.224,
0.225])



Построение моей модели ConvNet



Optuna



O P T U N A

- фреймворк, который оптимизирует гиперпараметры моделей.

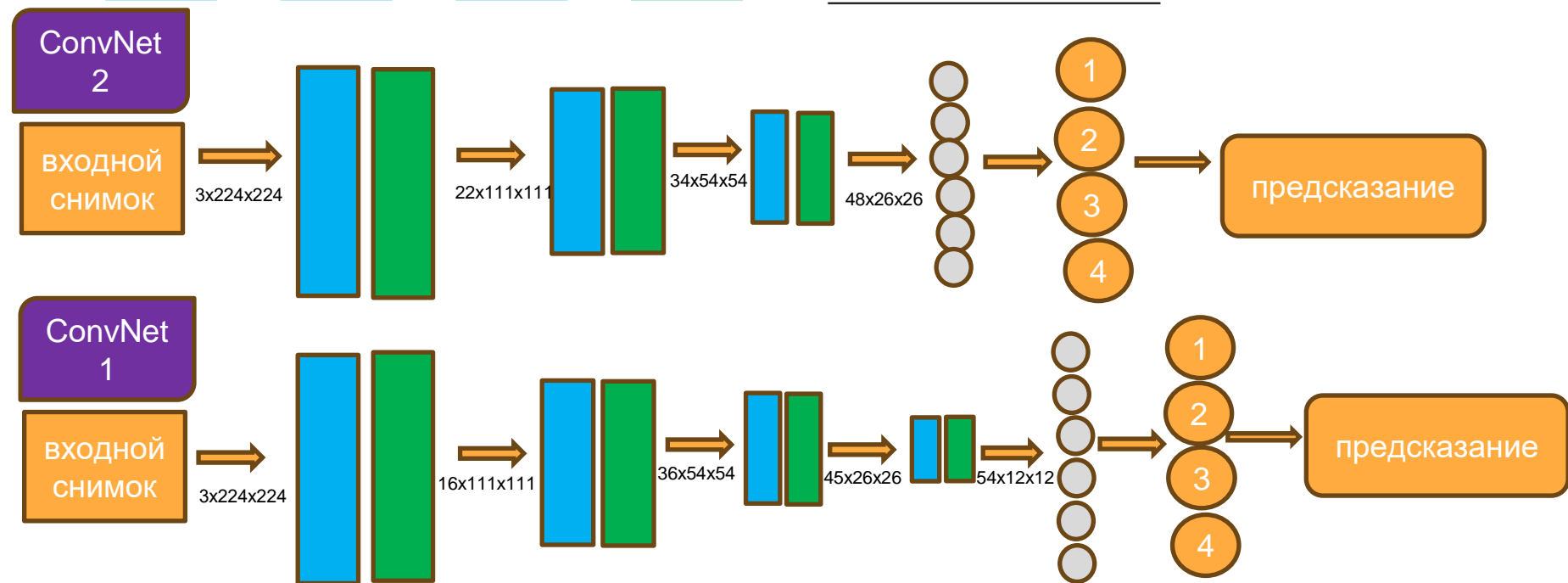
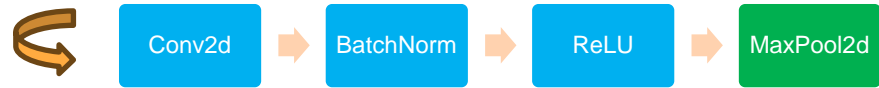
подбор наиболее эффективных комбинаций осуществляется с помощью алгоритма **Tree-structured Parzen Estimator (TPE)**

метрикой является целевая функция измерения качества модели, на каждой итерации осуществляется подбор на основе результатов предыдущих экспериментов

есть набор алгоритмов для прореживания экспериментов, которые позволяют отсекают варианты, которые провальные с большой вероятностью

№	Layer	Shape	Out channels
0	input	224x224	3
1	conv1	111x111	16
2	conv2	54x54	36
3	conv3	26x26	45
4	conv4	12x12	54

№	Layer	Shape	Out channels
0	input	224x224	3
1	conv1	111x111	22
2	conv2	54x54	34
3	conv3	26x26	48



Сравнение точности

69.66%

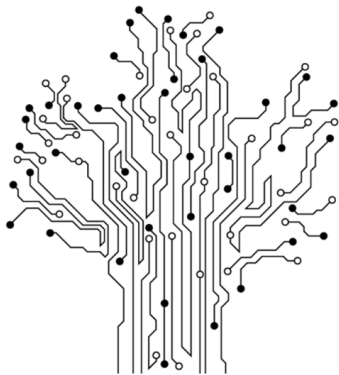
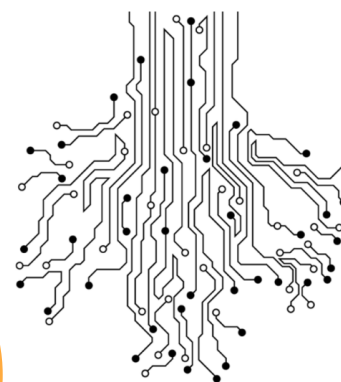
ConvNet1

66.33%

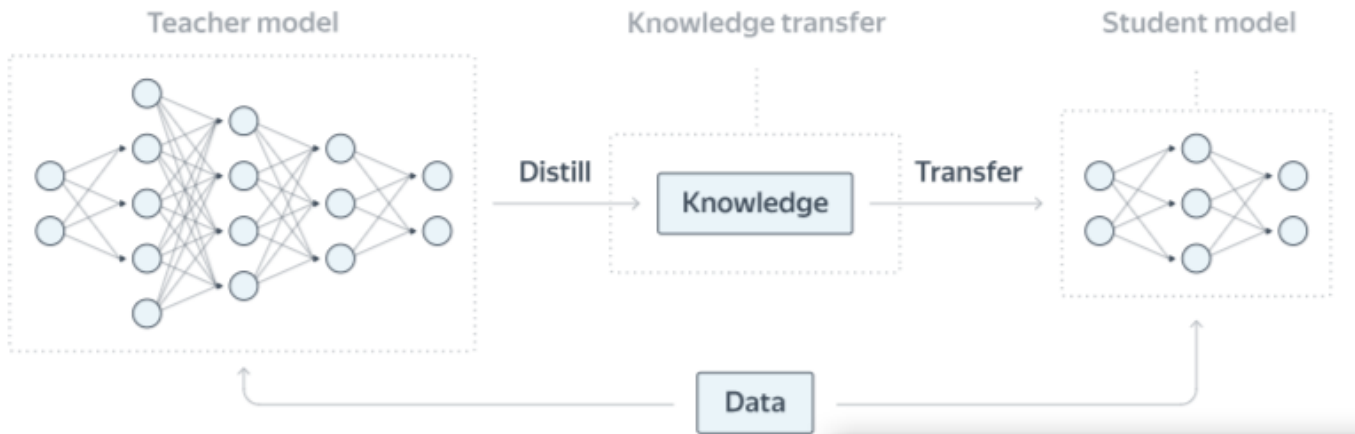
ConvNet2

70.61%

ConvNet



Дистилляция знаний



Дистилляция знаний
- это способ обучения нейросетей, направленный на передачу знаний от модели-учителя к модели-ученику.

Мы извлекаем дополнительную информацию из предсказания учителя для обучения ученика.

Во время обучения модели-ученика хотим добавлять дополнительное значение к функции потерь кросс-энтропии, на основе возвращаемых значений сети-учителя.

Дистилляция знаний

$$\text{loss} = \text{soft target loss weight} * \text{soft targets loss} + \text{ce loss weight} * \text{label loss}$$

- **soft target loss weight** - вес, присвоенный дополнительной цели, которую мы собираемся включить.
- **T** - температура, контролирует плавность выходных распределений. Чем больше значение T, тем более плавными становятся распределения, и тем больший прирост получают меньшие вероятности.
- **soft targets loss** = $\frac{\sum \text{targets} \cdot \text{prob}}{\text{classes}} T^2$
- **classes** - число классов классификации
- **targets, prob** - возвращаемые значения сети-учителя, к которым применены функции soft max и log soft max соответственно.

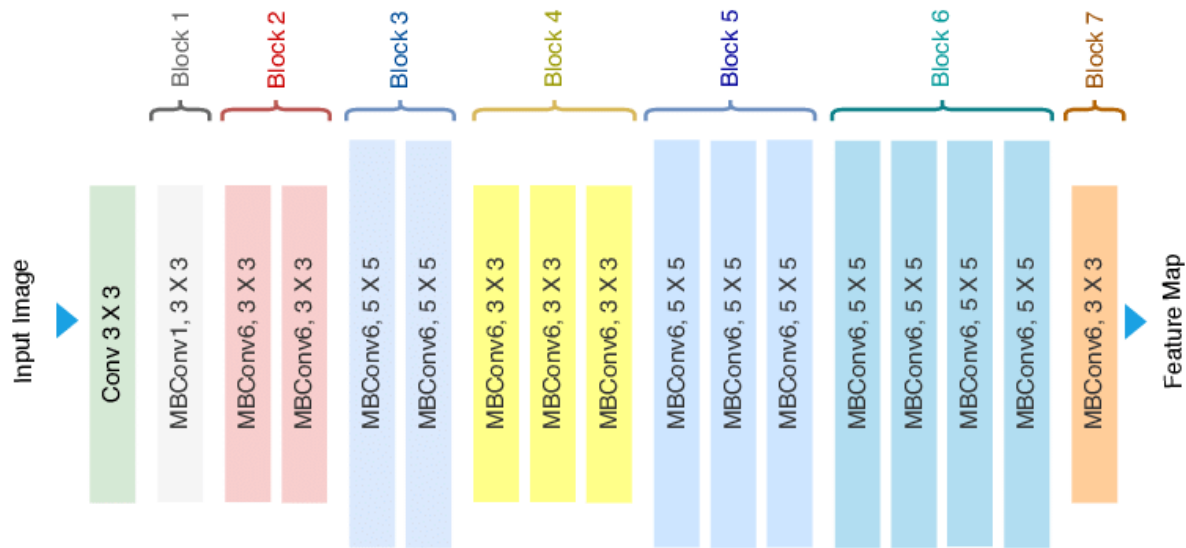
- **ce loss weight** - вес, присвоенный кросс-энтропии
- **label loss** - обычные потери кросс-энтропии, используемые при обучении студента

Метод из статьи:
Geoffrey Hinton, Oriol
Vinyals и Jeff Dean.
*Distilling the Knowledge
in a Neural Network*

Дистилляция знаний

В роли модели учителя выступит **EfficientNet-b0**, состоящая из 6,251,904 параметров.

Используем её предобученную на ImageNet, и дообучим на наших данных.



Дистилляция знаний

Модель	Accuracy OASIS	Accuracy Tumors
ConvNet(без учителя)	70.61%	68.53%
EfficientNet-b0	71.88%	72.08%
ConvNet(с учителем)	71.81%	73.10%

LRP

LRP (Layer-wise Relevance Propagation) — это функция интерпретации, который используется для определения важности каждого пикселя во входном изображении для прогнозирования выходного класса

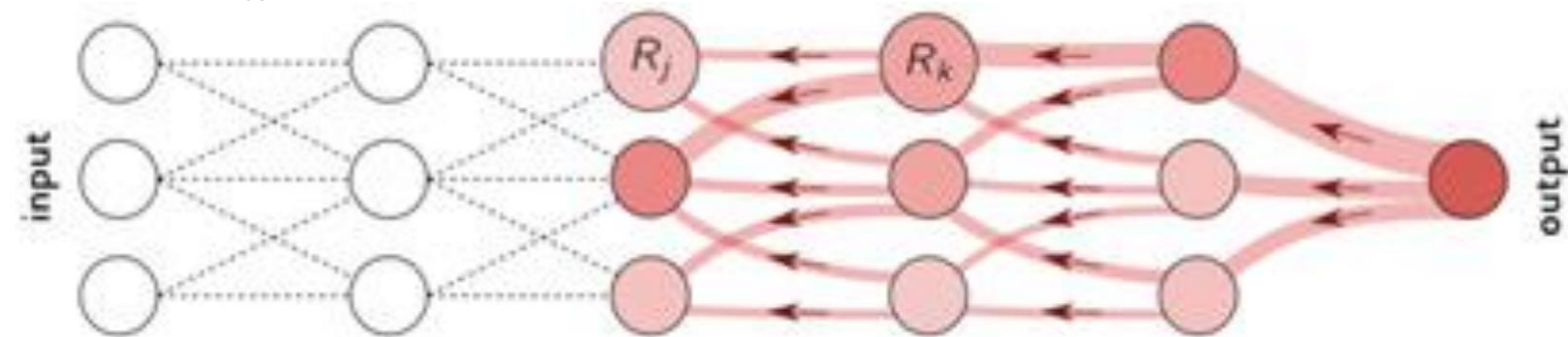
Epsilon-rule

$$R_j = \sum_k \frac{a_j w_{jk}}{\epsilon + \sum_{0,j} a_j w_{jk}} R_k$$

➡ R_j - важность нейрона j

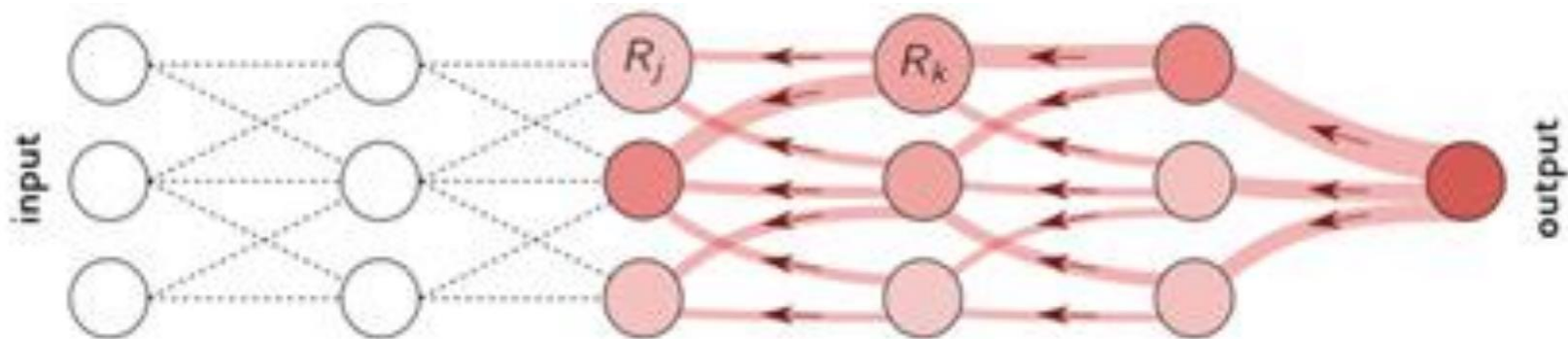
➡ $a_j w_{jk}$ - влияние нейрона j на нейрон k

➡ $a_k = \max(0, \sum a_j w_{jk})$

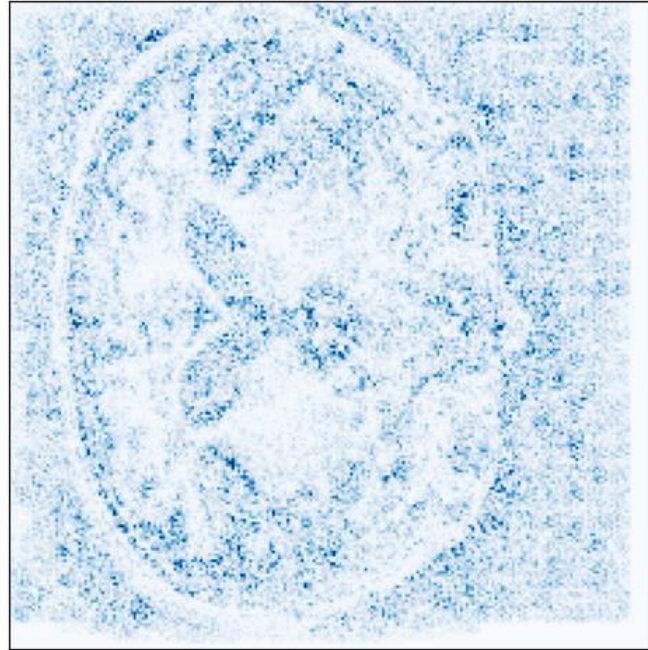
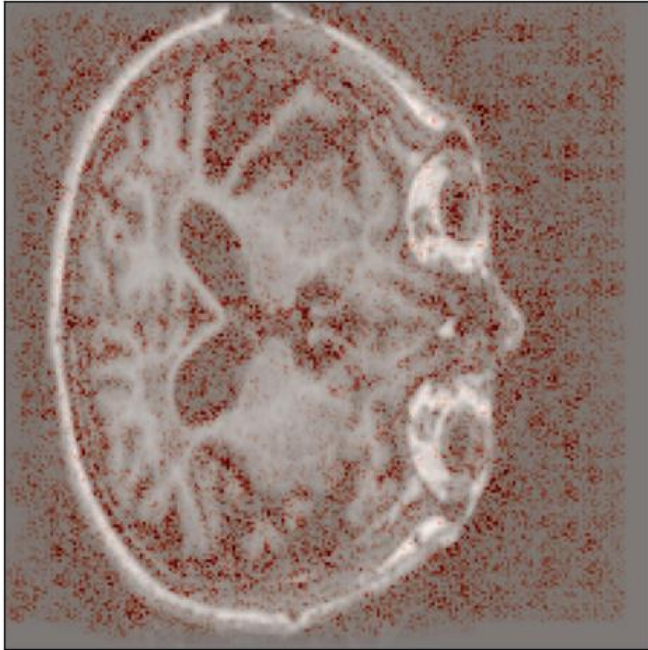


LRP

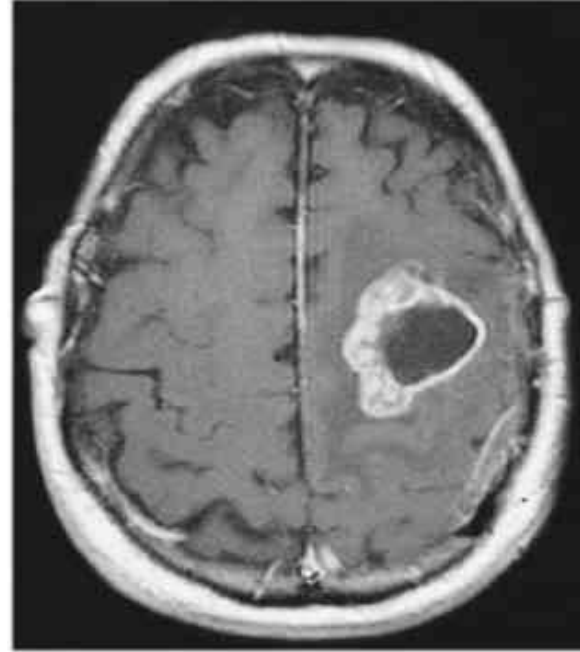
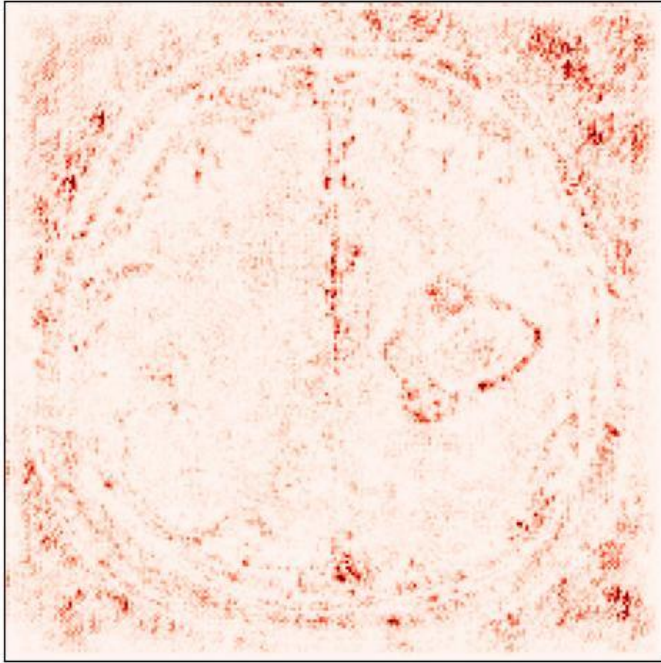
- forward pass - проход вперед - $\forall_k : z_k = \epsilon + \sum_j a_j \cdot w_{jk}$, можно вызвать метод forward предварительно применив правило lrp к весам
- element-wise division - поэлементное деление - $\forall_k : s_k = R_k / z_k$
- backward pass - обратный проход - $\forall_j : c_j = \sum_k w_{jk} \cdot s_k$ - можно применить backward и работаем с тензорами pytorch, можем вызвать метод grad
- element-wise product - поэлементное умножение - $\forall_j : R_j = a_j c_j$



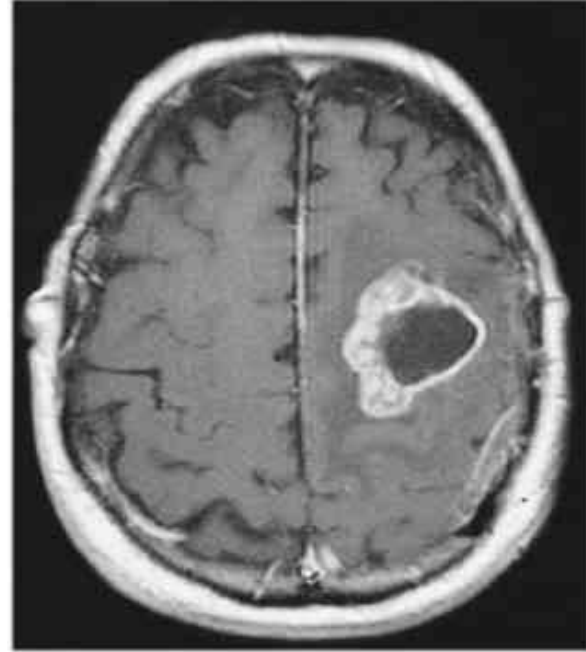
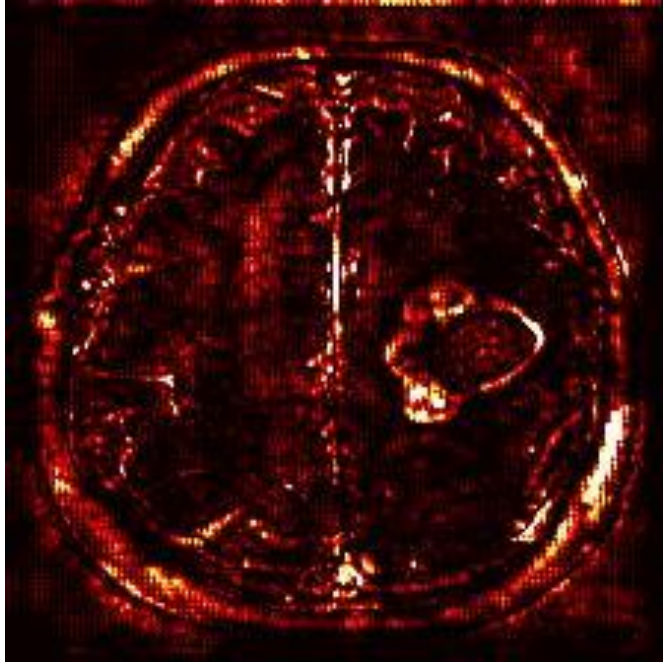
LRP by Captum



LRP by Captum



LRP by me



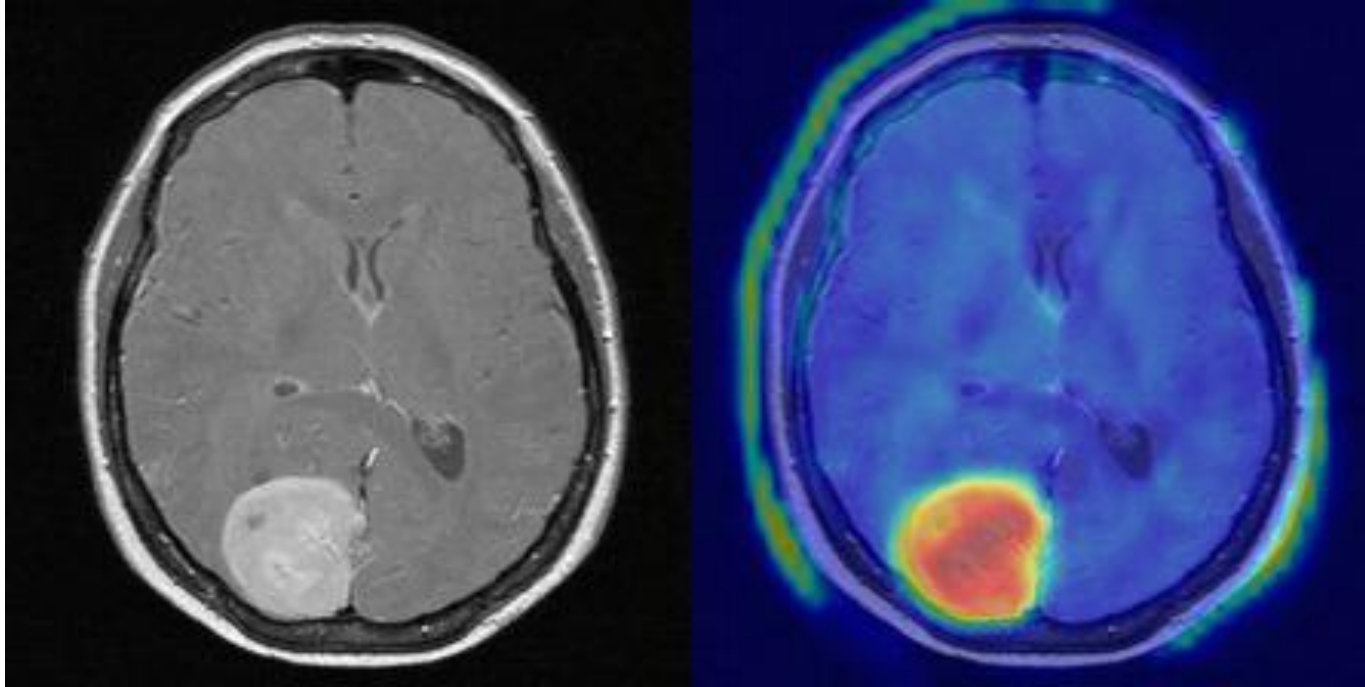
GRAD-CAM

Grad-CAM - Gradient-weighted Class Activation Mapping

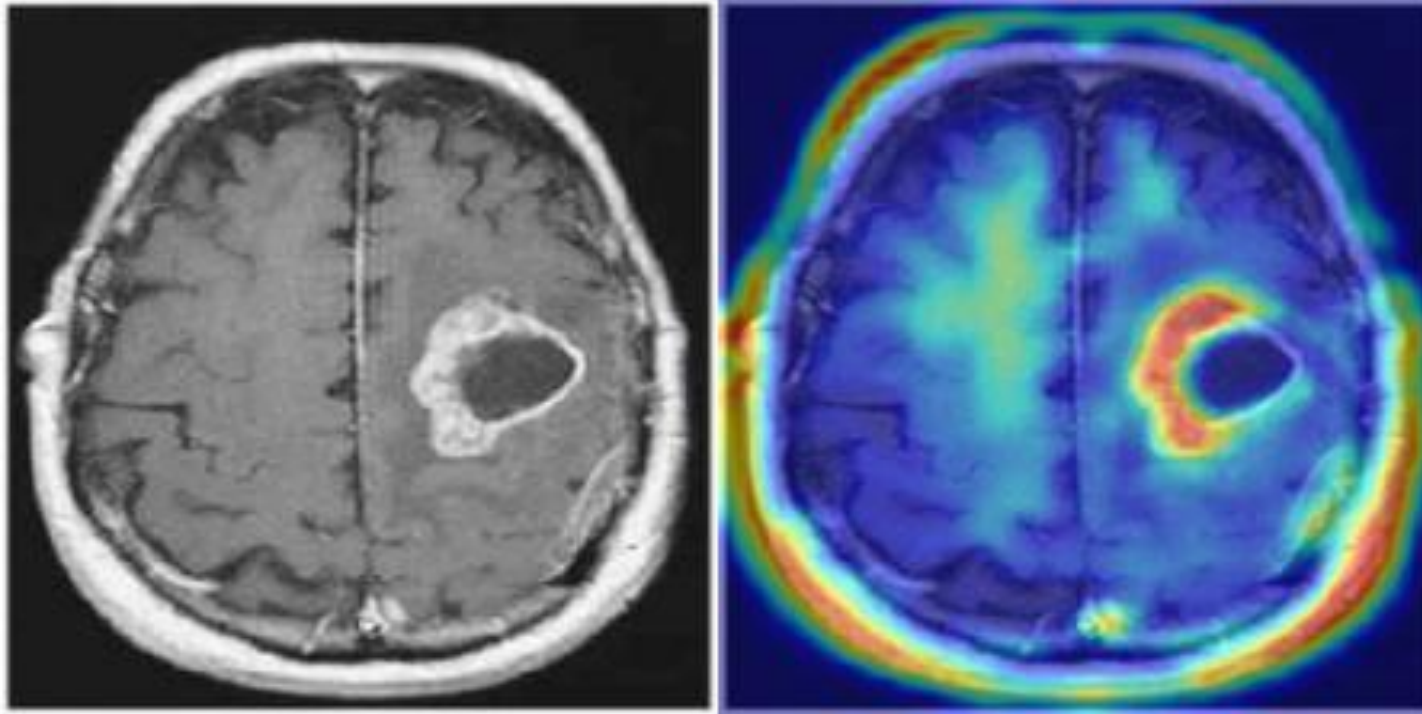
вычисляет градиенты модели относительно входного изображения, а затем использует для взвешивания активаций последнего свёрточного слоя.

- Подсчет градиента — $\frac{\partial Y^c}{\partial A_{ij}^k}$ — где Y^c — выходные данные для класса C до применения Softmax, и берем производную по картам признаков
- Подсчет весов усреднением градиентов - $w_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial Y^c}{\partial A_{ij}^k}$, где Z - константа равная числу пикселей в карте активации
- Подсчет Grad-CAM Heatmap - Relu примененная к линейной комбинации весов и карты признаков - $Relu(\sum_k w_k^c * A^k)$

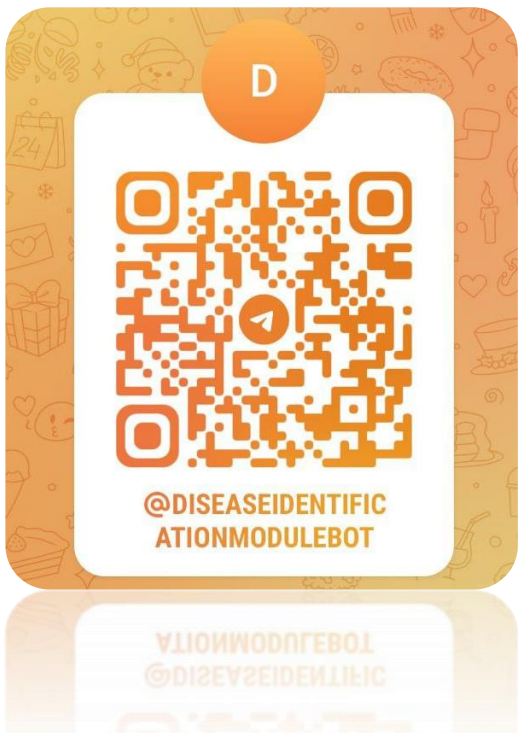
GRAD-CAM



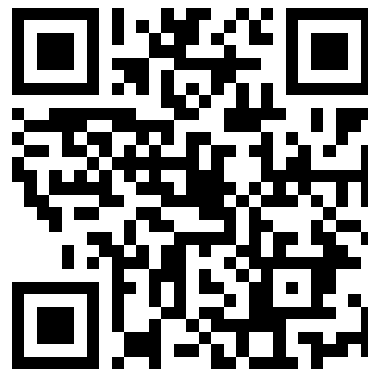
GRAD-CAM



Телеграмм-бот



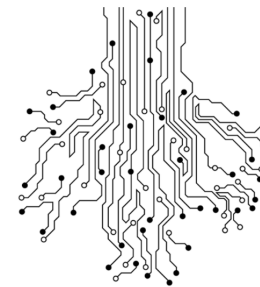
@DiseaseIdentificationModuleBot



Папка с картинками
для тестирования бота

<https://disk.yandex.ru/d/vTghYEzRhZRliQ>

Заключение



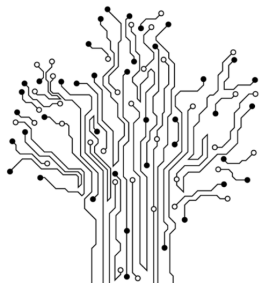
Создана модель, работающая с точностью **>70%** на обоих датасетах

Освоен фреймворк Optuna

Применена технология дистилляции знаний

Изучены и протестированы методы интерпретации, видим что на моем датасете Grad-Cam показывает более наглядную картину чем LRP

Создан телеграмм-бот **@DiseaseIdentificationModuleBot**





Спасибо за внимание!!!

