

Очистка от битых пикселей снимков и обучение автоэнкодера в рамках задачи распознавания туберкулеза на снимках с помощью сверточных нейронных сетей

Шигаров Никита 344 гр.
Научный руководитель
Невоструев Константин

Глобальная задача

- Конечная цель проекта — создание приложения, выявляющего туберкулез на флюорографиях легких с помощью сверточной нейронной сети

Этапы

- Удаление битых пикселей со снимка (Шигаров Н.)
- Определение, является ли кусок из снимка легким (Захаров Р.)
- Обучение автоэнкодера выделять важные признаки кусков снимков легких (Шигаров Н.)
- Использовать слои автоэнкодера в сверточной нейронной сети для распознавания туберкулеза

Подзадача №1. Битые пиксели.

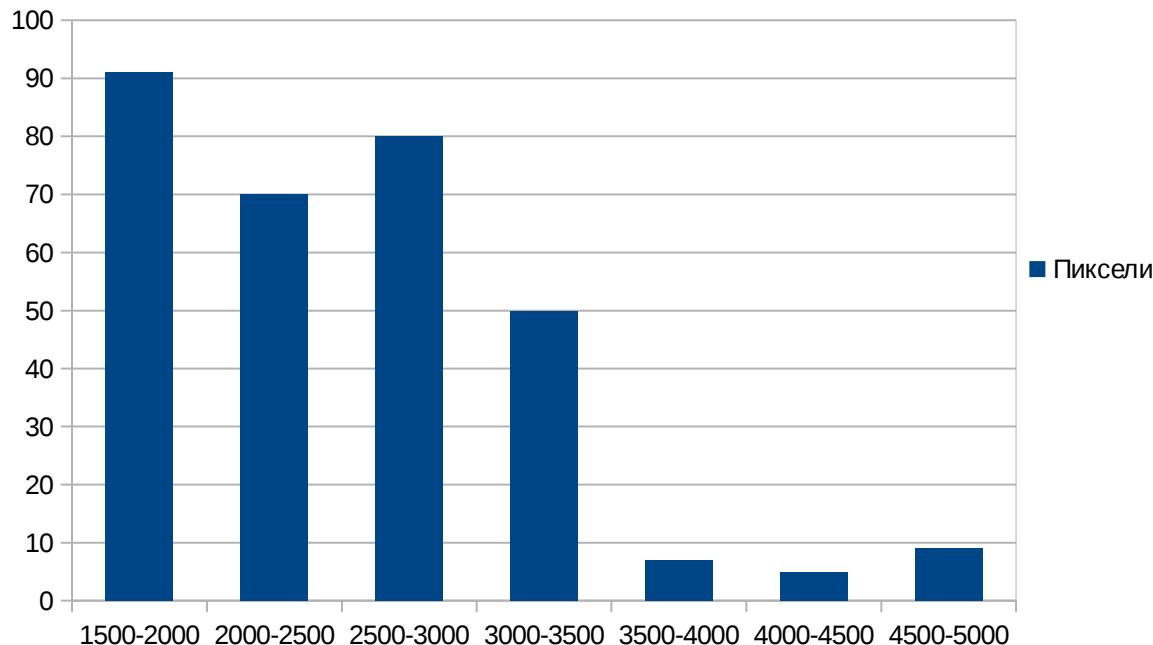
Существующие решения

- Медианный фильтр
- Усредняющий фильтр
- Гауссовый фильтр

- Python, библиотека scikit-image

Реализация на гистограммах

- Разработан свой метод для данной задачи
- Построение гистограмм на небольших участках (100 * 100)
- Нахождение выбросов с конца
- Метод будет срабатывать из-за большой разницы значений яркости пикселей в данном формате (макс. яркость 65536)

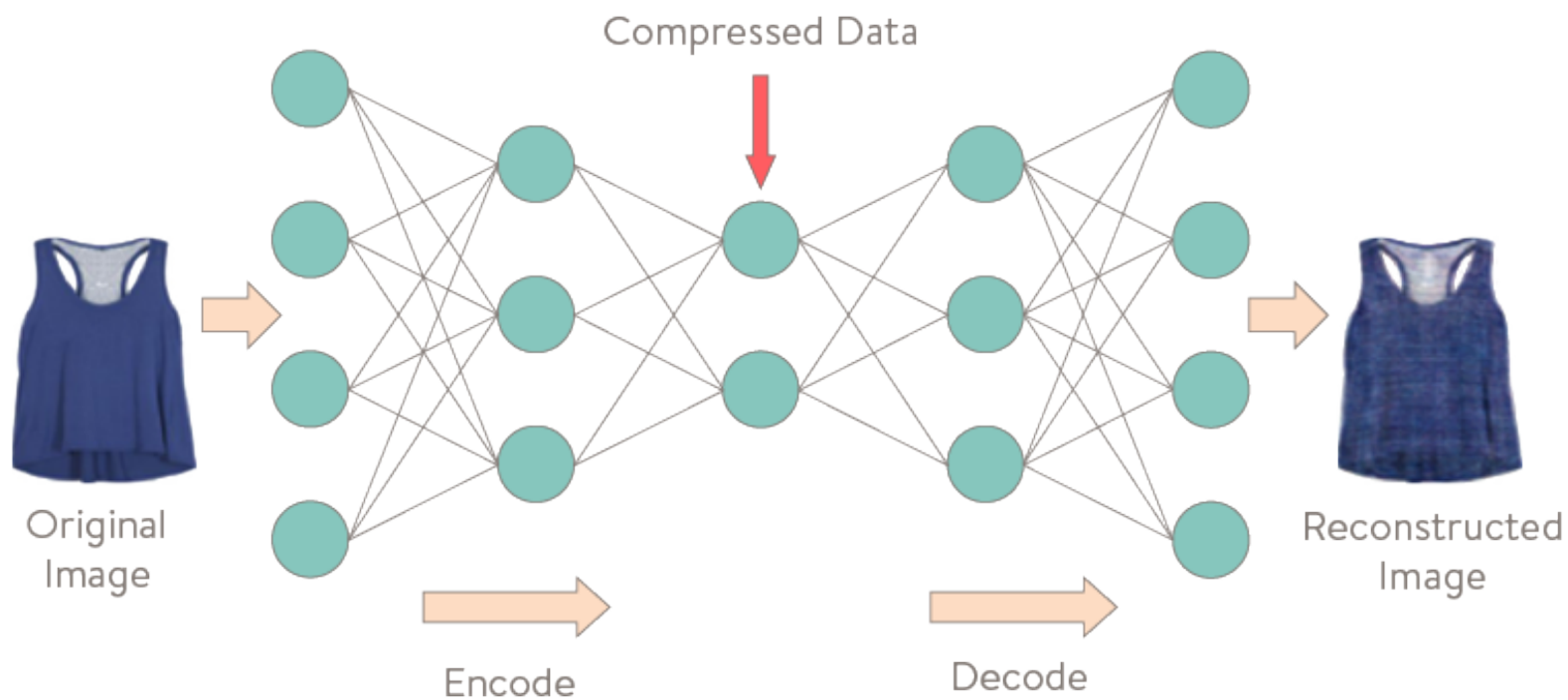
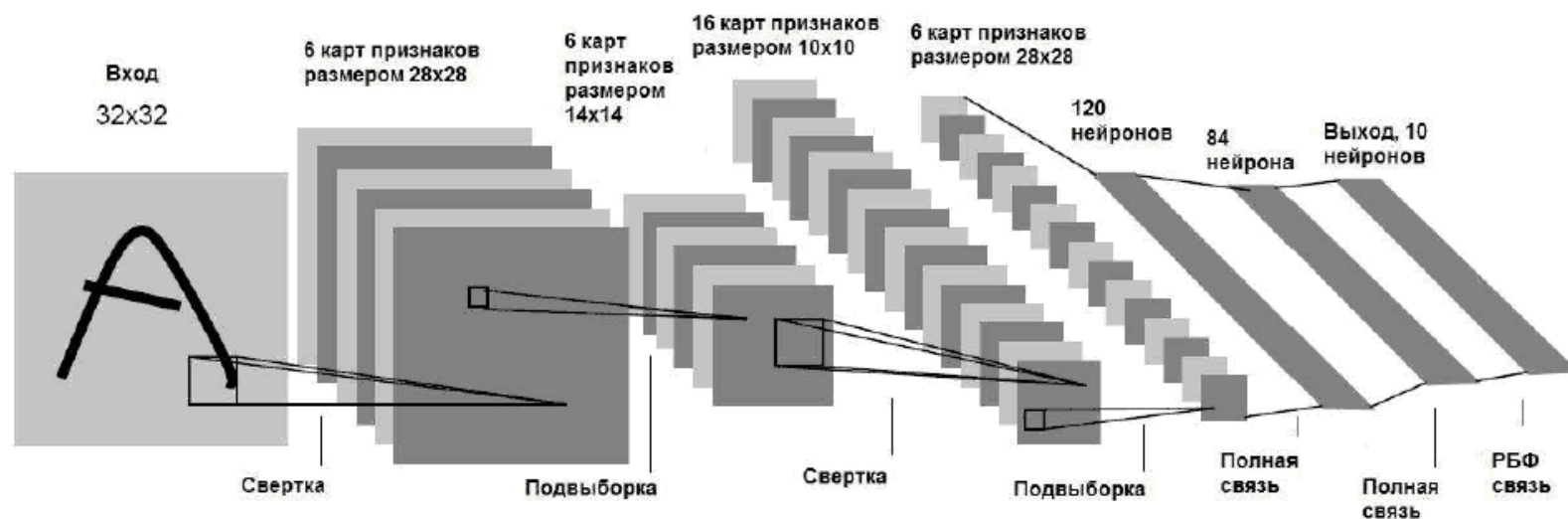


Сравнение

- Intel Core i7 4702MQ 2200 Mhz
- 1000 снимков, размер ядер в фильтрах: 5*5

Способ	Мат. ожидание	Среднеквадр. отклонение
Медианный фильтр	756 мс	102 мс
Усредняющий фильтр	523 мс	94 мс
Гауссовый фильтр	437 мс	73 мс
Гистограммы	174 мс	77 мс

Подзадача №2. Автоэнкодер



Программа для обучения

Обучать автоэнкодер будем послойно, с помощью своей программы на Python. На каждой итерации:

- Копируем веса из обученных внешних слоев
- Перебираем параметры внутреннего слоя (количество нейронов и размер окна)

В отдельном потоке:

- Делаем очистку снимков от битых пикселей
- Отправляем очищенные куски снимков в очередь

В основном:

- Берем очищенные куски снимков из очереди
- Обучаем с помощью библиотеки Caffe
- Находим наилучший вариант параметров текущего слоя по среднему значению евклидовой нормы разности входа и выхода на тестовой выборке

Результаты работы программы

- Обучение на 1000 кусках снимках легких
- Тестирование на 100 кусках снимках легких
- 5 замеров работы программы

Топология сети	Мат. ожидание времени работы	Среднеквадр. отклонение	Средняя норма разности входа и выхода
Conv – Pool – Conv – Pool	658 сек	86 сек	110.45
Conv – Conv – Pool	1034 сек	114 сек	75.67
Conv – Pool - Conv	511 сек	62 сек	87.45

Результаты

- Реализован собственный алгоритм по нахождению битых пикселей на снимках легких, скорость которого в 3-4 раза выше стандартных решений удаления шума на изображениях
- Создано многопоточное приложение, очищающее снимки от битых пикселей и обучающее послойно автоэнкодер с разными параметрами слоев, выбирая наилучший вариант