

Федеральное государственное бюджетное образовательное учреждение
высшего образования «Санкт-Петербургский государственный
университет»

Кафедра Системного Программирования

Шигаров Никита Алексеевич

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

Курсовая работа

Научный руководитель:
ведущий разработчик ООО "НМТ" Невоструев К.Н.

Санкт-Петербург
2016

Оглавление

1. Введение	3
1.1. Мотивация	3
1.2. Постановка задачи	3
1.3. Доступные программные средства	4
2. Битые пиксели	5
2.1. Описание проблемы	5
2.2. Существующие решения	5
2.3. Способ с использованием гистограмм	6
2.4. Сравнение производительности	7
3. Обучение автоэнкодера	9
3.1. Описание проблемы	9
3.2. Сверточные нейронные сети и автоэнкодер	9
3.3. Послойное обучение	10
3.4. Программа для обучения	11
3.5. Результаты работы программы	12
4. Результаты	13
Список литературы	14

1. Введение

1.1. Мотивация

Туберкулез — это социально-опасная проблема и нужно уметь ее обнаруживать на ранней стадии. Часто врачи загружены работой, а специалистов, которые могут проанализировать флюорографии, обычно мало. Идея состоит в том, чтобы сделать для них систему помощи, которая может отсеивать снимки, на которых патологии точно нет.

Лучшим вариантом распознавания изображений на текущий момент являются сверточные нейронные сети, об этом можно судить по многочисленным статьям [3] и соревнованию ImageNet¹, где всегда используются сверточные нейронные сети. Однако, нужно уметь правильно их настраивать, чтобы добиться хорошего результата.

1.2. Постановка задачи

Весь проект можно разделить на подзадачи:

- Очистить снимки от битых пикселей
- По случайно вырезанным кускам симков научиться определять их принадлежность легким
- Обучить автоэнкодер на таких кусках с помощью своей автоматической системы для улучшения обучения сверточной нейронной сети
- Обучить сверточную нейронную сеть, используя слои автоэнкодера

В данной курсовой предлагаются решения 1 и 3 задачи.

¹Домашняя страница соревнования ImageNet LSVRC 2015, URL: <http://image-net.org/challenges/LSVRC/2015/> (дата обращения: 22.05.2016)

1.3. Доступные программные средства

При вычислениях использовались:

- Язык программирования Python 2.7² — высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода
- Библиотека NumPy³ — предоставляет реализации вычислительных алгоритмов (в виде функций и операторов), оптимизированные для работы с многомерными массивами
- Библиотека Scikit-image⁴ — включает в себя алгоритмы сегментации, геометрических преобразований, манипуляции цветового пространства, анализ, фильтрацию изображений и многое другое
- Библиотека Caffe⁵ — реализует алгоритмы глубокого машинного обучения, имеет хорошую документацию, большое сообщество, удобна в использовании

²Домашняя страница языка Python, URL: <https://www.python.org/> (дата обращения: 22.05.2016)

³Домашняя страница библиотеки NumPy, URL: <http://www.numpy.org/> (дата обращения: 22.05.2016)

⁴Домашняя страница библиотеки Scikit-image, URL: <http://scikit-image.org/> (дата обращения: 22.05.2016)

⁵Домашняя страница библиотеки машинного обучения Caffe, URL: <http://caffe.berkeleyvision.org/> (дата обращения: 22.05.2016)

2. Битые пиксели

2.1. Описание проблемы

В рамках данной задачи мы имеем дело не с обычными картинками, а со специальным форматом медицинских снимков — DICOM. Пиксели кодируются 65536 оттенками серого. Но, как и на обычной картинке, могут встречаться скопления битых пикселей, в следствие загрязнения объектива медицинского аппарата. Нужно иметь быстрый и корректный алгоритм их удаления для того, чтобы нейронная сеть не распознала их как отклонения.

2.2. Существующие решения

Стандартным способом удаления шума на изображениях является использование медианного, усредняющего или гауссового фильтров.

Фильтры применялись к копиям снимков, а затем производилось сравнение исходного и исправленного снимка. Там, где пиксели отличались на некоторую константу, производилась замена на среднее по соседним не битым пикселям.

Медианный фильтр — метод нелинейной обработки сигналов, разработанный Тьюки. Двумерный медианный фильтр представляет собой скользящее окно $N \times N$, охватывающее нечетное число элементов изображения. Центральный элемент заменяется медианой всех элементов в окне. Медианой дискретной последовательности для нечетного K является тот ее элемент, для которого существуют $\frac{K-1}{2}$ элементов, меньших или равных ему по величине, и $\frac{K-1}{2}$ элементов, больших или равных ему по величине. Медианный фильтр работает хорошо в рамках данной задачи: при достаточном размере окна битые пиксели не влияют на соседние, что приводит к корректному удалению битых пикселей и даже пятен.

Усредняющий фильтр — метод заключается в замене значения яркости в текущем пикселе на среднюю яркость, вычисленную по его 8 окрестностям, включая и сам элемент. Этот фильтр является самым

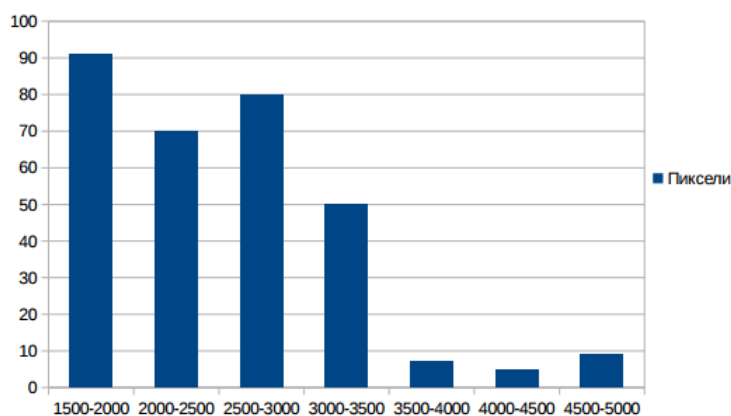
простым. Данный фильтр будет работать хуже медианного из-за того, что битые пиксели, имеющие существенно отличное значение яркости будут вносить значительный вклад в результат обработки, что повлияет на нахождение лишних пикселей, которые будут восприняты как битые.

В гауссовом фильтре матрица заполняется по нормальному (гауссовому) распределению. После чего применяется операция свертки к изображению. Данная матрица образует такое взвешенное среднее, что коэффициенты дальних пикселей будут меньше центральных. Данный метод будет работать точнее усредняющего в случаях, когда битые пиксели находятся на границе матрицы и будут меньше влиять на результат. Однако в случаях, когда битые пиксели, или, что еще хуже пятна, находятся ближе к центру, мы будем существенно изменять значение не битого пикселя.

2.3. Способ с использованием гистограмм

Для данной задачи был придуман свой алгоритм нахождения битых пикселей. Идея заключается в построении гистограмм на небольших участках снимков и нахождении выбросов на них. Так, мы можем двигаться от самых ярких в сторону менее ярких интервалов до тех пор, пока не зафиксируем интервал с достаточным количеством пикселей. Пройденные интервалы можно считать интервалами, содержащими битые пиксели, которые заменяются на среднее по соседним.

Способ будет срабатывать из-за большого количества оттенков в данном формате, когда значения битых пикселей будут сильно отличаться от значений пикселей на рассматриваемом участке. Таким образом, битые пиксели будут попадать в другие интервалы с более высокими значениями.



Для избежания проблемы ложного определения битых пикселей, когда светлый участок снимка попал на границу рассматриваемого участка, используются перекрытия. То есть новый участок будет захватывать границу предыдущего и, если пиксель на границе, воспринятый как битый в предыдущем участке, не будет входить в битый интервал, то его не будем считать битым. Таким образом, данный способ будет работать так же корректно, как и медианный.

2.4. Сравнение производительности

- Было выполнено 1000 измерений на процессоре Intel Core i7 4702MQ 2200 Mhz на различных флюорографиях легких одинакового формата
- Размер ядер медианного, гауссового и усредняющий фильтров: 5 * 5
- Язык Python, фильтры из библиотеки scikit-image

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

Как можно видеть, способ с использованием гистограмм в данной задаче в 3-4 раза быстрее стандартных вариантов удаления шума на изображениях.

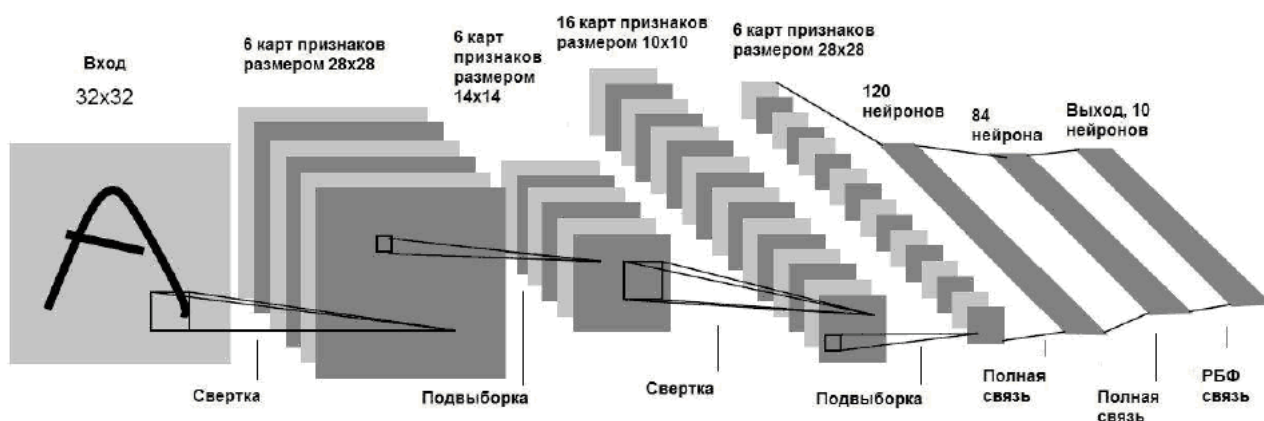
3. Обучение автоэнкодера

3.1. Описание проблемы

В следствие отсутствия большого количества размеченных данных — кусков снимков, отмеченных наличием патологий, нужно воспользоваться предварительной настройкой слоев сверточной нейронной сети для повышения точности распознавания.

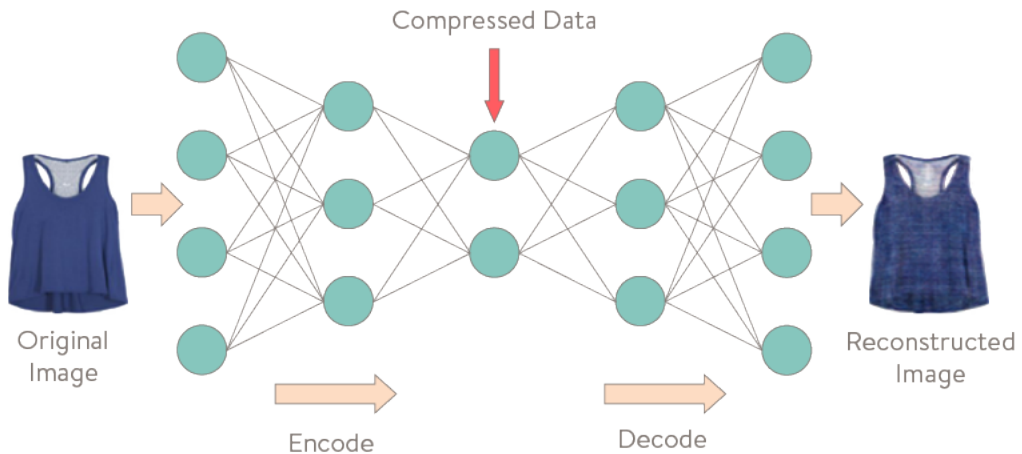
3.2. Сверточные нейронные сети и автоэнкодер

Как известно, слои сверточной нейронной сети представляют собой сверточные и подвыборочные. На выходе их работы предполагаются значения, представляющие характерные признаки изображений (в нашем случае куски снимков). [4] В конце установим полносвязные слои, представляющие собой классификатор, входом которого будут признаки снимков, а выходом значение, отвечающее за наличие патологии, его обучим с помощью метода обратного распространения ошибки. [5]



Обучать сверточные и подвыборочные слои предлагается отдельно с помощью автоэнкодера. Это специальная архитектура искусственных нейронных сетей, позволяющая применять обучение без учителя при использовании метода обратного распространения ошибки. Автоэнкодер стремится получить на выходе то же, что и на входе. Его слои делятся по типу на encoding, состоящих из сверточных и подвыборочных

и decoding, состоящих из слоев, обратных к сверточным и подвыборочным. [1]



Поскольку размерность изображения снижается после подвыборочных слоев, автоэнкодер учится выделять характерные черты изображений, чтобы по ним восстановить вход. Таким образом, после обучения encoding слои и будут представлять собой сверточные и подвыборочные слои нашей сверточной нейронной сети.

3.3. Послойное обучение

В статье *Bengio et al* [2] дается объяснение почему послойный способ обучения будет наиболее эффективным.

Этот метод обучения заключается в следующем. Сначала обучим автоэнкодер с двумя парами encoding и decoding слоев (сверточный и подвыборочный в каждой части) с помощью стандартного метода обратного распространения ошибки, предварительно задав количество нейронов и размер матриц в сверточных и подвыборочных слоях. Затем создадим новый автоэнкодер к которому будет добавлено аналогично по паре внутренних encoding и decoding слоев с другой конфигурацией. Потом скопируем веса внешних слоев из предыдущей обученной модели и запустим обучение новой, отключив при этом обучение внешних слоев и так далее.

3.4. Программа для обучения

Для автоматизации послойного обучения была реализована многопоточная программа на Python, работа которой заключается в следующем. 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 с помощью метода с использованием гистограмм
- Отправляем очищенные куски снимков в очередь

В основном:

- Берем очищенные куски снимков из очереди
- Обучаем с помощью библиотеки Caffe

После обучения:

- Находим наилучший вариант параметров текущего слоя по среднему значению евклидовой нормы разности входа и выхода на тестовой выборке
- Переходим к следующей итерации

3.5. Результаты работы программы

- Обучение на 1000 кусках легких
- Тестирование на 100 кусках 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

Как можно видеть, время обучения на первой топологии сети было существенно меньше, чем на второй из-за того, что в ней присутствуют подвыборочный слой, между сверточными, в следствие чего обучение второго сверточного слоя будет происходить быстрее из-за меньшего размера изображения.

Зато вторая сеть дает более точный результат из-за того, что изображение сжимается всего один раз и можно сохранить больше информации о его признаках.

4. Результаты

В рамках данной работы были достигнуты следующие результаты:

- Изучена литература и статьи о сверточных нейронных сетях.
- 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

Список литературы

- [1] Baldi Pierre. Autoencoders, Unsupervised Learning, and Deep Architectures. — Department of Computer Science, University of California, Irvine, 2012.
- [2] Bengio Yoshua, Courville Aaron, Vincent Pascal. Representation Learning: A Review and New Perspectives. — Department of computer science and operations research, U. Montreal, Canada, 2013. P. 13-16.
- [3] Bengio Yoshua, LeCun Yann. Convolutional Networks for Images, Speech and Time-Series. — MIT Press Cambridge, MA, USA, 1998. P. 255-258.
- [4] Xu Jun, Luo Xiaofei, Wang Guanhao et al. A Deep Convolutional Neural Network for segmenting and classifying epithelial and stromal regions in histopathological images. — Neurocomputing 191, P. 214–223, USA, 2016. P. 215-216.
- [5] Хайкин С. Нейронные сети: Полный курс. Пер. с англ. Н. Н. Кускуль, А. Ю. Шелестова. 2-е изд., испр. — М.: Издательский дом Вильямс, 2008, 1103 с.