

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное автономное образовательное  
учреждение высшего образования  
ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ

Институт математики, механики и компьютерных наук  
имени И. И. Воровича

Направление подготовки  
Фундаментальная информатика и информационные технологии

А. С. Коваленко

ОБУЧЕНИЕ ШУМОПОДАВЛЯЮЩЕЙ НЕЙРОННОЙ СЕТИ БЕЗ  
ИСПОЛЬЗОВАНИЯ ЧИСТЫХ ДАННЫХ

Магистерская диссертация

Научный руководитель:  
учёная степень, учёное звание / должность И. О. Фамилия  
Рецензент:  
ученая степень, ученое звание, должность И. О. Фамилия

Ростов-на-Дону  
2020

# ОГЛАВЛЕНИЕ

Введение	3
0.1. Постановка задачи . . . . .	6
0.2. Несколько примеров в $\text{\LaTeX}$ . . . . .	8
0.2.1. Как вставлять листинги и рисунки . . . . .	8
0.2.2. Как оформить таблицу . . . . .	11
0.2.3. Как набирать формулы . . . . .	11
0.2.4. Как оформлять списки . . . . .	11
Заключение	13
Список литературы	14

## ВВЕДЕНИЕ

Шумоподавление является часто встречаемой задачей в области компьютерного зрения. Так как любое изображение, полученное из картины реального мира является дискретным представлением непрерывного аналогового сигнала, то в нём будет присутствовать шум. Изначально шум появляется у полезного сигнала из-за погрешностей приёма оптического излучения фотоидами, данное явление изложено в книге *Fundamentals of linear electronics* [1]. Затем к данному искажённому сигналу добавляются потери при процессе дискретизации. Действие шума на изображения можно легко увидеть на примере изображения 0.1, на правой части изображения приводится пример фотографии, снятой при более благоприятных условиях, при которых фотосенсор цифровой камеры порождает меньше шума, чем на фотографии слева.

Также существуют изображения, полученные не только с фотосенсоров. Примером можно привести рентгенографию, широко используемую во многих областях, таких как медицина, процессы производства и эксплуатации, криминалистика, реставрация и экспертиза художественных ценностей. Существует два варианта получения цифрового изображения рентгенографии, это оцифровка уже существующего рентгеновского снимка, и использование технологии цифровой рентгенографии, при которой сразу идет цифровая обработка получаемого рентгеновского изображения. Оба данных метода подвержены наличию шума на получаемом цифровом изображении, в большей мере шум преобладает на изображениях, получаемых первым способом, так как идёт дополнительное наложение шума сканиру-



Рисунок 0.1 — Пример зашумленного изображения.

ющим устройством. Пример подобного шума проиллюстрирован на изображении 0.2. Аналогичная ситуация наблюдается и с снимками в области радиографического контроля сварных соединений, пример проиллюстрирован на изображении 0.3.

Помимо рентгенографии в обработке медицинских изображений потребность в избавлении полезного сигнала от шума встречаются и в других методах диагностики и визуализации, таких как цифровая реконструкция компьютерной томографии, магнитно-резонансной томографии и других. Потребность в шумоподавлении на медицинских цифровых изображениях изложена в работе [2].

На данный момент с развитием технологий глубокого обучения и построения глубоких архитектур сверточных нейронных сетей, такие архитектуры применяются к решению широкого ряда задач в области обработки и анализа изображений. Существует три подхода к обучению нейронных сетей: обучение с учителем, обучение без учителя и обучение с подкреплением. Если рассматривать близкие задачи к шумоподавлению с точки зрения построения шаблона обуче-

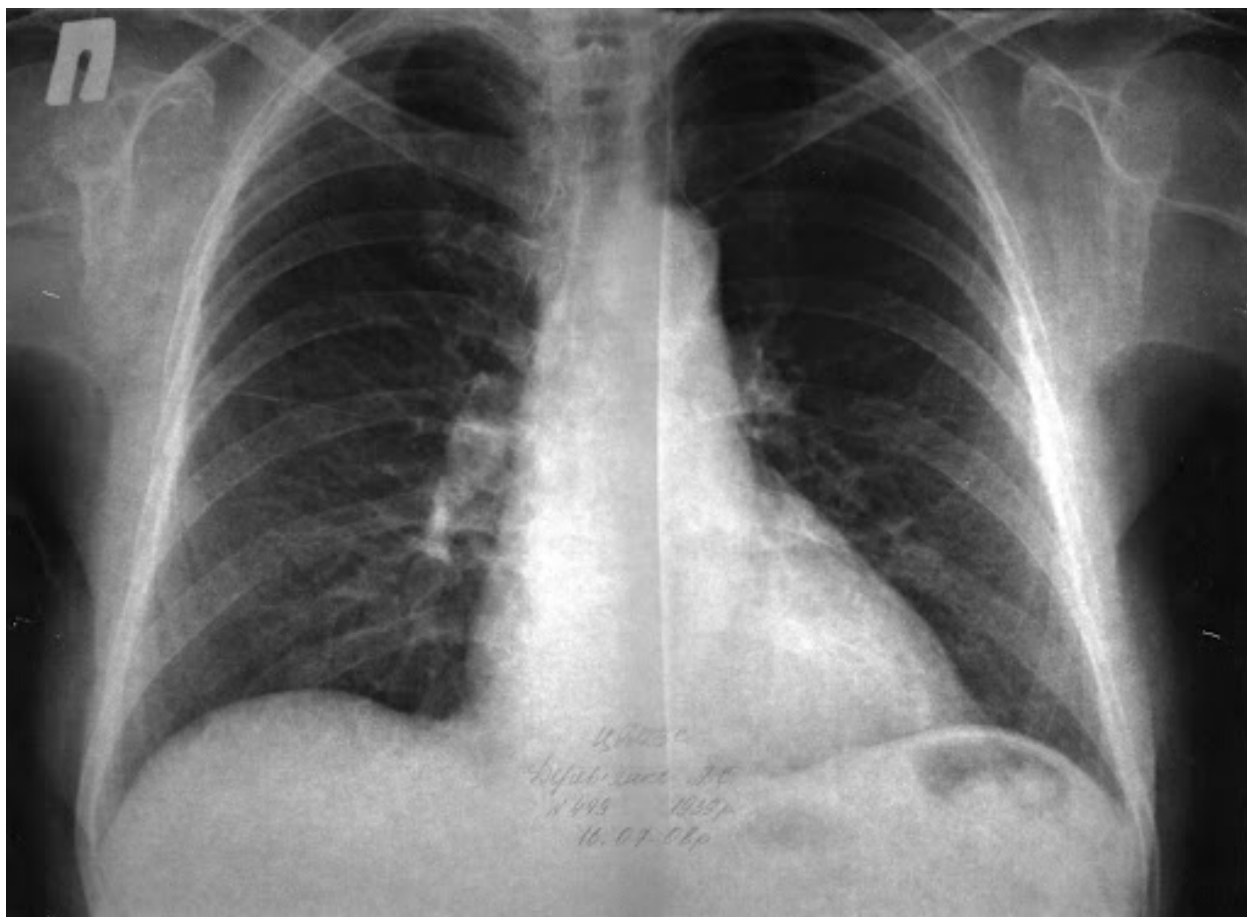


Рисунок 0.2 — Сканированный рентгеновский снимок грудной клетки человека.

ния сети, это задача увеличения разрешения изображения с помощью нейронных сетей [3] и задача восстановления изображения [4]. Для решения данных задач нейросетевые архитектуры обучаются с помощью подхода обучения с учителем [5]. При процессе обучения таких архитектур в качестве входных данных выступают сжатые варианты изображений, подаваемых, как ожидаемые при предсказании сети. Если рассматривать в данном контексте обучение для задачи классификации, то на вход сети необходимо подавать изображение с присутствующим на нём шумом, и обучать сеть предсказывать уже само чистое изображение без шума.

Все перечисленные способы в этом разделе получения цифрового изображения объединяет один недостаток, это невозможность

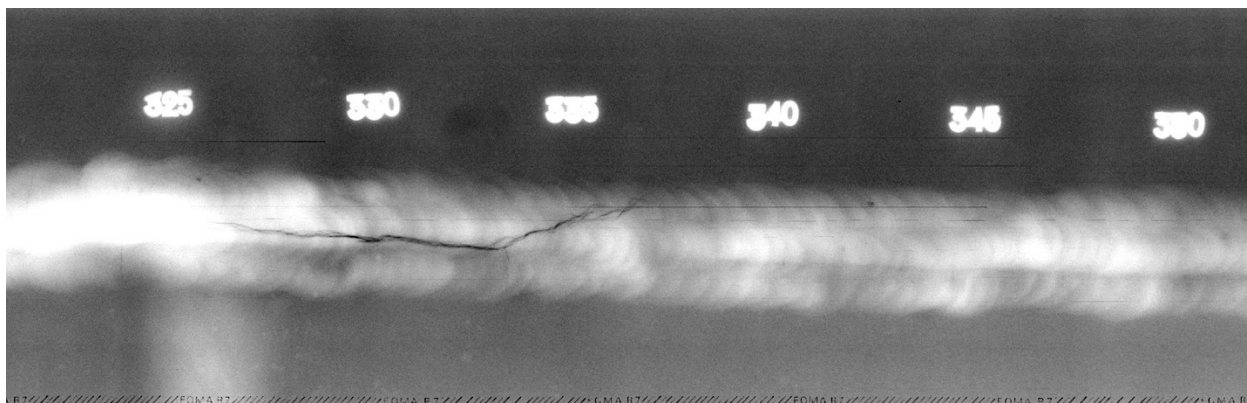


Рисунок 0.3 — Цифровой рентгеновский снимок сварного соединения.

получения чистого изображения без шума для обучения шумоподавляющей нейронной сети. В данной работе рассматривается подход к построению архитектуры шумоподавляющей нейронной сети и построения обучающего процесса, основанного на методе обучения без учителя (неконтролируемое обучение).

## 0.1. Постановка задачи

Пусть имеется изображение в формате непрерывного сигнала, полученного с АЦП об сигналах матрицы фотосенсора фиксированного цифрового устройства в момент времени  $t$  (DNG формат [6]):

$$S(t) \tag{0.1}$$

Сигнал  $S(t)$  0.1 состоит из полезного сигнала  $G(t)$  и шума  $r(t)$ , порождаемым фотосенсором:

$$S(t) = G(t) + r(t) \tag{0.2}$$

При применении преобразования сырого сигнала  $S(t)$  к трехмерному дискретному изображению в цветовой схеме RGB получаем

матрицу изображения  $\tilde{I}$ , с влиянием шума квантования при округлении сигнала при его дискретизации. Подход преобразования  $p$  описан в работе Processing RAW Images in MATLAB [7].

$$\tilde{I}_{i,j} = p(S(t))_{i,j} + q, \quad q \sim \mathcal{N}(-\frac{1}{2}, \frac{1}{12}), \quad (0.3)$$

где  $q$ , это шум квантования, семплируемый из нормального распределения с параметрами  $\mathcal{N}(-\frac{Q}{2}, \frac{Q}{12})$  при шаге квантования  $Q = 1$ , более подробно можно ознакомиться с этим в главе Оценки ошибок (шумов) квантования выходного сигнала в цифровом фильтре из книги Цифровая обработка сигналов [8].

Для упрощения постановки задачи будем полагать, что матрица изображения  $\tilde{I}$  состоит из суммы изображения, полученного из преобразования чистого (полезного) сигнала  $G(t)$  0.2 в матрицу  $I$  и шума  $\alpha$ , полученного из случайного распределения  $\mathbb{P}$ , так как появление шума  $r(t)$  0.3 можно считать абсолютно случайным процессом.

$$\tilde{I} = I + \alpha, \quad \alpha \sim \mathbb{P} \quad (0.4)$$

Таким образом, имея серию из  $N$  изображений одной и той же сцены, снятых в разный момент времени  $t$  имеем следующую выборку:

$$\{\tilde{I}_k\}_{k=1}^N, \text{ где } \tilde{I}_k = I + \alpha_k, \quad \alpha_k \sim \mathbb{P} \quad (0.5)$$

Причём для каждого элемента из набора  $\tilde{I}_k$  компонента  $I$  является одним и тем же значением, так как при съёмке одной сцены полезный сигнал  $G(t)$  0.2 остаётся неизменным.

Целью данной работы является построение с помощью нейронной сети приближения отображения  $\phi : \mathbb{R}^n \rightarrow \mathbb{R}^n$ , обладающим следующим свойством:

$$\forall \tilde{I} \in \{\tilde{I}_k\}_{k=1}^N \implies \phi(\tilde{I}) = I \quad (0.6)$$

Для построения приближения отображения  $\phi$ , нейронная сеть  $f$  будет обучаться решать следующую задачу оптимизации:

$$\min_w \|f(\tilde{I}) - I\|_{L_2}, \quad (0.7)$$

где  $w$  - параметры сети  $f$ ,  $L_2$  - евклидова норма [9].

## 0.2. Несколько примеров в $\text{\LaTeX}$

Некоторые часто используемые команды приведены в качестве примера ниже (и варианты — в комментариях). Мы рекомендуем внимательно прочесть данный текст и изучить его исходный код прежде, чем начинать писать свой собственный. Кроме того, можно дать и такой совет: идущий ниже текст не убирать до самого конца, а просто оставлять его позади своего собственного текста, чтобы в любой момент можно было проконсультироваться с данными примерами.

### 0.2.1. Как вставлять листинги и рисунки

Для крупных листингов есть два способа. Первый красивый, но в нём не допускается кириллица (у вас может встречаться в комментариях и печатаемых сообщениях), он представлен на листинге 0.2.1.

---

#### Листинг 0.2.1. Программа “Hello, world” на C++

---

```
#include <iostream>
using namespace std;

int main()
{
    cout << "Hello, world" << endl;
    system("pause");
    return 0;
}
```

---



Второй не такой красивый, но без ограничений (см. листинг 0.2.2).

---

**Листинг 0.2.2.** Программа “Hello, world” без подсветки

---

```
#include <iostream>
using namespace std;

int main()
{
    cout << "Привет, мир" << endl;
}
```

---

Можно использовать первый для вставки небольших фрагментов внутри текста, а второй для вставки полного кода в приложении, если таковое имеется.

Если нужно вставить совсем короткий пример кода (одна или две строки), то выделение линейками и нумерация может смотреться чересчур громоздко. В таких случаях можно использовать окружения `lstlisting` или `Verb` без `ListingEnv`. Приведём такой пример с указанием языка программирования, отличного от заданного по умолчанию:

```
fibs = 0 : 1 : zipWith (+) fibs (tail fibs)
```

Такое решение — со вставкой нумерованных листингов покрупнее и вставок без выделения для маленьких фрагментов — выбрано, например, в книге Эндрю Таненбаума и Тодда Остина по архитектуре компьютера [10] (см. рис. 0.4).

Наконец, для оформления идентификаторов внутри строк (функция `main` и тому подобное) используется `lstinline` или, самое простое, моноширинный текст (`\texttt`).

Использовать внешние файлы (например, рисунки) можно и на [overleaf.com](https://overleaf.com): ищите кнопку `upload`.

После вызова этой процедуры решение должно выводиться на экран. Сначала процедура проверяет, равно ли единице значение  $n$ . Если да, то решение тривиально: нужно просто переместить один диск с  $i$  на  $j$ . Если  $n$  не равно 1, решение состоит из трех частей и каждая из этих частей представляет собой рекурсивную процедуру.

Все решение представлено в листинге 5.6. Рассмотрим такой вызов процедуры:

```
towers (3, 1, 3)
```

Этот вызов порождает еще три вызова:

```
towers (2, 1, 2)
```

```
towers (1, 1, 3)
```

```
towers (2, 2, 3)
```

Первый и третий вызов производят по три вызова каждый, и всего получится семь.

**Листинг 5.6.** Процедура для решения задачи «Ханойская башня»

```
public void towers (int n, int i, int j) {  
    int k;  
    if (n == 1)  
        System.out.println("Переместить диск с " + i + " на " + j);  
    else {  
        k=6-i-j;  
        towers(n-1, i, k);  
        towers (1, i, j);  
        towers (n-1, k, j);  
    }  
}
```

Для рекурсивных процедур нам нужен стек, чтобы, как и в JVM, хранить параметры и локальные переменные каждого вызова. Каждый раз при вызове процедуры на вершине стека располагается новый стековый кадр для процедуры. Текущий кадр — это кадр, созданный последним. В наших примерах стек растёт

Рисунок 0.4 — Пример оформления листингов в [10]

Таблица 0.1. — Подпись к таблице — сверху

Item		
Животное	Описание	Цена (\$)
Gnat	per gram	13.65
	each	0.01
Gnu	stuffed	92.50
Emu	stuffed	33.33
Armadillo	frozen	8.99

### 0.2.2. Как оформить таблицу

Для таблиц обычно используются окружения `table` и `tabular` — см. таблицу 0.1. Внутри окружения `tabular` используются специальные команды пакета `booktabs` — они очень красивые; самое главное: использование вертикальных линеек считается моветоном.

### 0.2.3. Как набирать формулы

$\text{\LaTeX}$  is great at typesetting mathematics. Let  $X_1, X_2, \dots, X_n$  be a sequence of independent and identically distributed random variables with  $E[X_i] = \mu$  and  $\text{Var}[X_i] = \sigma^2 < \infty$ , and let

$$S_n = \frac{X_1 + X_2 + \dots + X_n}{n} = \frac{1}{n} \sum_i^n X_i$$

denote their mean. Then as  $n$  approaches infinity, the random variables  $\sqrt{n}(S_n - \mu)$  converge in distribution to a normal  $\mathcal{N}(0, \sigma^2)$ .

### 0.2.4. Как оформлять списки

Нумерованные списки (окружение `enumerate`, команды `item`)...

1. Like this,

2. and like this.

...маркированные списки ...

- Like this,
- and like this.

...списки-описания ...

**Word** Definition

**Concept** Explanation

**Idea** Text

## ЗАКЛЮЧЕНИЕ

Помните, что на все пункты списка литературы должны быть ссылки.  $\text{\LaTeX}$  просто не добавит информацию об издании из bib-файла, если на это издание нет ссылки в тексте. Часто студенты используют в работе электронные ресурсы: в этом нет ничего зазорного при одном условии: при каждом заимствовании следует ставить соответствующую ссылку. В качестве примера приведём ссылку на сайт нашего института [11].

Для дальнейшего изучения  $\text{\LaTeX}$  рекомендуем книгу Львовского [12]: она хорошо написана, хотя и несколько устарела. Обычно стоит искать подсказки на [tex.stackexchange.com](http://tex.stackexchange.com), а также читать документацию по установленным пакетам с помощью команды

```
texdoc имя_пакета
```

или на [ctan.org](http://ctan.org).

## СПИСОК ЛИТЕРАТУРЫ

1. *Cox J.* Fundamentals of linear electronics: integrated and discrete. — US : DELMAR, 2001.
2. *Авшаров Е. М., Абгарян М. Г., Сафарянц С. А.* Обработка медицинских изображений как необходимый инструментарий медицинского диагностического процесса // Вестник рентгенологии и радиологии. — 2010. — № 3. — С. 54—61.
3. Image Super-Resolution Using Very Deep Residual Channel Attention Networks / Y. Zhang [и др.] // CoRR. — 2018. — Т. abs/1807.02758. — arXiv: 1807 . 02758. — URL: <http://arxiv.org/abs/1807.02758>.
4. *You S., You N., Pan M.* PI-REC: Progressive Image Reconstruction Network With Edge and Color Domain // CoRR. — 2019. — Т. abs/1903.10146. — arXiv: 1903 . 10146. — URL: <http://arxiv.org/abs/1903.10146>.
5. *Nasteski V.* An overview of the supervised machine learning methods // HORIZONS.B. — 2017. — Дек. — Т. 4. — С. 51—62. — DOI: 10.20544/HORIZONS.B.04.1.17.P05.
6. *Adobe.* Digital negative (DNG) Specification. — US, 2012. — URL: [https://www.adobe.com/content/dam/acom/en/products/photoshop/pdfs/dng\\_spec\\_1.4.0.0.pdf](https://www.adobe.com/content/dam/acom/en/products/photoshop/pdfs/dng_spec_1.4.0.0.pdf).
7. *Sumner R.* Processing RAW Images in MATLAB. — 2014. — URL: [https://rcsumner.net/raw\\_guide/RAWguide.pdf](https://rcsumner.net/raw_guide/RAWguide.pdf).

8. Гондельберг Л. М., Матюшкин Б. Д., Поляк И. Н. Цифровая обработка сигналов. — Москва : Радио и связь, 1985. — URL: <http://optic.cs.nstu.ru/files/Lit/Math/gold.pdf>.
9. Хайкин С. Нейронные сети : Полный курс. — Москва : Вильямс, 2016.
10. Таненбаум Э., Остин Т. Архитектура компьютера. — 6-е изд. — СПб. : Питер, 2013.
11. Сайт Мехмата ЮФУ. — URL: <http://mmcs.sfedu.ru> (дата обр. 01.01.2015).
12. Львовский С. М. Набор и вёрстка в системе  $\text{\LaTeX}$ . — М. : МЦНМО, 2006. — URL: <http://www.mccme.ru/free-books/llang/newllang.pdf>.