

Санкт-Петербургский Государственный Университет  
Факультет Прикладной Математики Процессов Управления

Основная образовательная программа:  
ВМ.5504 Исследование операций и системный анализ  
Направление: 01.04.02 Прикладная математика и информатика

Авдеенко Дмитрий Юрьевич

# Обработка изображений с помощью методов машинного обучения

Выпускная квалификационная работа

Научный руководитель:  
к. т. н., доцент Гришкин В. М.

Рецензент:  
ст. научный сотрудник  
ООО "Нокиа солюшнз энд нетворкс"  
Епифинов Н. А.

Санкт-Петербург  
2020

# Оглавление

<b>Введение</b>	<b>3</b>
<b>Постановка задачи</b>	<b>4</b>
<b>Обзор литературы</b>	<b>5</b>
<b>1. Обзор технологий и подходов в решении задачи</b>	<b>6</b>
1.1. U-Net . . . . .	6
1.2. SENet . . . . .	7
1.3. PSPNet . . . . .	8
1.4. Rectified Adam оптимизация . . . . .	10
1.5. Коэффициент Жаккара . . . . .	12
1.6. Псевдо-маркировка данных . . . . .	12
<b>2. Данные</b>	<b>13</b>
<b>3. Модель решения</b>	<b>18</b>
<b>4. Исследование</b>	<b>21</b>
4.1. Подготовка данных и инструментария для исследования	21
4.2. Многоклассовая классификация . . . . .	21
4.3. Многоклассовая сегментация . . . . .	23
4.4. Бинарная сегментация . . . . .	25
4.5. Результаты . . . . .	28
<b>Выводы</b>	<b>29</b>
<b>Заключение</b>	<b>30</b>
<b>Список литературы</b>	<b>31</b>

# Введение

Сталь является одним из важнейших строительных материалов современности. Стальные изделия устойчивы к естественному и искусственному износам, что поспособствовало популярности данного материала во всем мире. Компания Северсталь лидирует в сфере добычи и производства стали и считает, что будущее металлургии требует развития в экономических, экологических и социальных аспектах отрасли, а также серьезно относится к корпоративной ответственности. Недавно компания создала крупнейшее в стране хранилище промышленных данных с петабайтами информации, которая ранее никак не изучалась. В настоящее время Северсталь ищет возможности в области машинного обучения для улучшения автоматизации, повышения эффективности и поддержания высокого качества своей продукции.

Процесс производства листовой стали особенно деликатный. От нагрева и прокатки до сушки и резки - к моменту готовности листа стали несколько машин соприкасаются с ним. Сегодня Северсталь использует изображения с высокочастотных камер для алгоритма обнаружения дефектов.

Основной идеей данной работы было создание алгоритма для автоматического анализа, локализации и классификации поверхностных дефектов на стальном листе.

# Постановка задачи

Цель этого исследования - предсказать местоположение и тип дефектов, обнаруженных при производстве стали, используя предоставленные изображения. Названия изображений имеют уникальный идентификатор, и задача состоит в том, чтобы сегментировать каждое изображение и классифицировать дефекты в наборе тестовых данных.

Даны обучающая выборка состоящая из 12568 уникальных изображений и тестовая выборка, состоящая из 1801 изображения. Так же предлагается текстовый файл, с информацией о дефекте на каждом изображении.

Каждое изображение может не иметь дефектов, иметь дефект одного класса или дефекты нескольких классов (всего представлено четыре различных класса).

Исследование состоит из двух последовательных задач:

1. Классификация дефекта на изображении в один из четырех классов.
2. Предсказать местонахождение найденных дефектов или сегментировать дефект.

Стоит отметить, что сегмент для каждого класса дефектов будет закодирован в одну строку, даже если на изображении имеется несколько несмежных местоположений дефектов.

Качество решения оценивается по среднему коэффициенту Сёренса (Dice coefficient). Коэффициент можно использовать для попиксельного сравнения предсказанной сегментации и соответствующей ей истинностью. Формула имеет вид:

$$\frac{2|X \cap Y|}{|X| + |Y|} \quad (1)$$

где  $X$  и  $Y$  предсказанный и истинный сектор на изображении соответственно.

# Обзор литературы

В процессе написания данной работы было изучено множество книг и статей по машинному обучению, нейронным сетям и распознаванию изображений.

Общие теоретические знания из области глубокого обучения и нейронных сетей, а также из математическое обоснование получены из книги [3].

В [2] подробно изложены методы и подходы для аугментации данных, что позволяет увеличить обучающую выборку и улучшить качество модели. Работа [1] содержит пример использования аугментации во время теста (test-time augmentation) для повышения качества модели сегментации, построенной на свёрточной нейронной сети.

В работе [8] представлен новый архитектурный элемент для свёрточных нейронных сетей Squeeze-and-Excitation(SE) блок, который помогает перекалибровать отклики по каналам путем явного моделирования взаимозависимостей между ними. Модели на основе нейронных сетей с такой архитектурой помогли занять первое место на конкурсе ILSVRC 2017, что улучшило результат прошлого года на 25%.

Различные методы оптимизации позволяют существенно улучшить скорость и качество обучения моделей. Авторами [6] представлены теоретические и практические результаты метода Rectified Adam (RAdam), путем исправления дисперсии для адаптивной скорости обучения, что в свою очередь является улучшением метода Adam [4]

# 1. Обзор технологий и подходов в решении задачи

Решение исходной задачи можно разделить на две компоненты - сегментирующую и классифицирующую составляющие. Основным способом решения задач классификации и сегментации изображений являются методы глубокого обучения. Нейронные сети являются основным инструментом решения задач такого рода. В настоящее время представлено большое количество различных архитектур нейронных сетей. Ниже описаны различные подходы, которые были использованы в исследовании.

## 1.1. U-Net

U-Net — это свёрточная нейронная сеть, которая изначально создана [7] и применена для сегментации биомедицинских изображений. Архитектура сети представляет собой полносвязную свёрточную сеть, модифицированную так, чтобы она могла работать с меньшим количеством примеров (обучающих образов) и делала более точную сегментацию.

Сеть содержит сжимающий путь (слева) и расширяющий путь (справа), поэтому архитектура похожа на букву U (Рис. 1), что и отражено в названии. На каждом шаге мы удваиваем количество каналов признаков.

Сжимающий путь похож на типичную свёрточную сеть, он содержит два подряд свёрточных слоя  $3 \times 3$ , после которых идет слой ReLU и пулинг с функцией максимума  $2 \times 2$  с шагом 2.

Каждый шаг расширяющего пути содержит слой, обратный пулингу, который расширяет карту признаков, после которого следует свёртка  $2 \times 2$ , которая уменьшает количество каналов признаков. После идет конкатенация с соответствующим образом обрезанной картой признаков из сжимающего пути и две свертки  $3 \times 3$ , после каждой из которой идет ReLU. Обрезка нужна из-за того, что мы теряем пограничные

пиксели в каждой свёртке. На последнем слое свертка  $1 \times 1$  используется для приведения каждого 64-компонентного вектора признаков до требуемого количества классов.

Всего сеть имеет 23 свёрточных слоя.

Данная модель до сих пор показывает хорошие результаты в области сегментирования изображений.

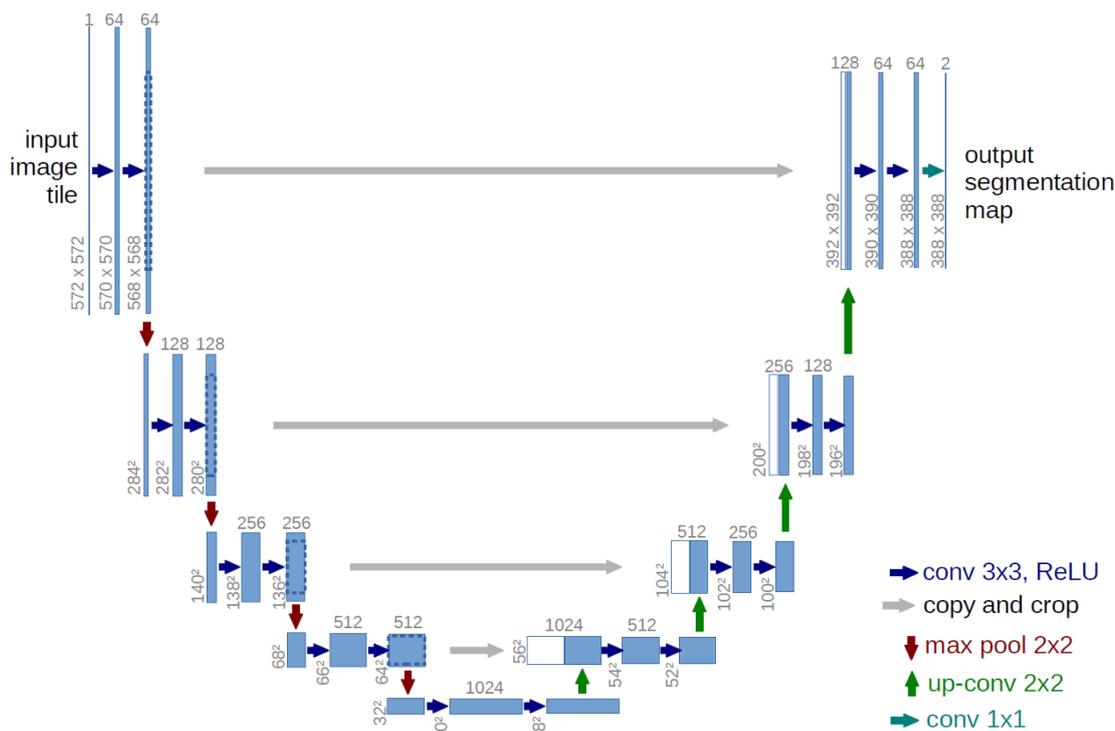


Рис. 1: Архитектура Unet

## 1.2. SENet

Squeeze-and-Excitation Networks (SENeTs) [8] представляют собой специальный блок для свёрточной нейронной сети, который улучшает взаимозависимости каналов практически без дополнительных вычислений. Нейронные сети такого типа позволили выиграть в конкурсе ImageNet [5] в 2017 году и с их помощью был улучшен результат прошлого года на 25%. Основная идея - добавить параметры к каждому каналу свёрточного блока, чтобы сеть могла адаптивно регулировать вес каждой карты признаков. Основным преимуществом данного подхода является

то, что SE-блоки могут быть легко добавлены к существующим архитектурам.

Имплементацию данного подхода можно описать с помощью следующей последовательности действий:

1. SENet блок получает входной свёрточный блок и текущее количество каналов.
2. Мы сжимаем каждый канал в одно числовое значение, используя усредненный пулинг.
3. Полносвязный слой, сопровождаемый функцией ReLU, добавляет необходимую нелинейность. Сложность выходного канала также уменьшается на определенное соотношение.
4. Второй полносвязный слой, за которым следует активация сигмоида, дает каждому каналу плавную пропускающую функцию.
5. Взвешиваем каждую карту признаков свёрточного блока на основе результата нашей основной нейронной сети.

Авторы оригинального исследования демонстрируют, что добавляя SE-блоки в ResNet-50(Рис. 2), мы можем ожидать почти такую же точность, как и у ResNet-101, что является существенным улучшением модели, так как требует только половину вычислительных затрат при сравнимом уровне качества. В статье также рассматриваются другие архитектуры, такие как Inception, Inception-ResNet и ResNeXt.

### 1.3. PSPNet

На рис. 3 изображена архитектура PSPNet.

На (а) расположено входное изображение. На (b) ResNet используется с расширенной сетевой стратегией для извлечения функций. Размер карты признаков здесь составляет  $1/8$  от входного изображения.

На (с) среднее объединение субрегионов выполняется для каждой карты объектов.

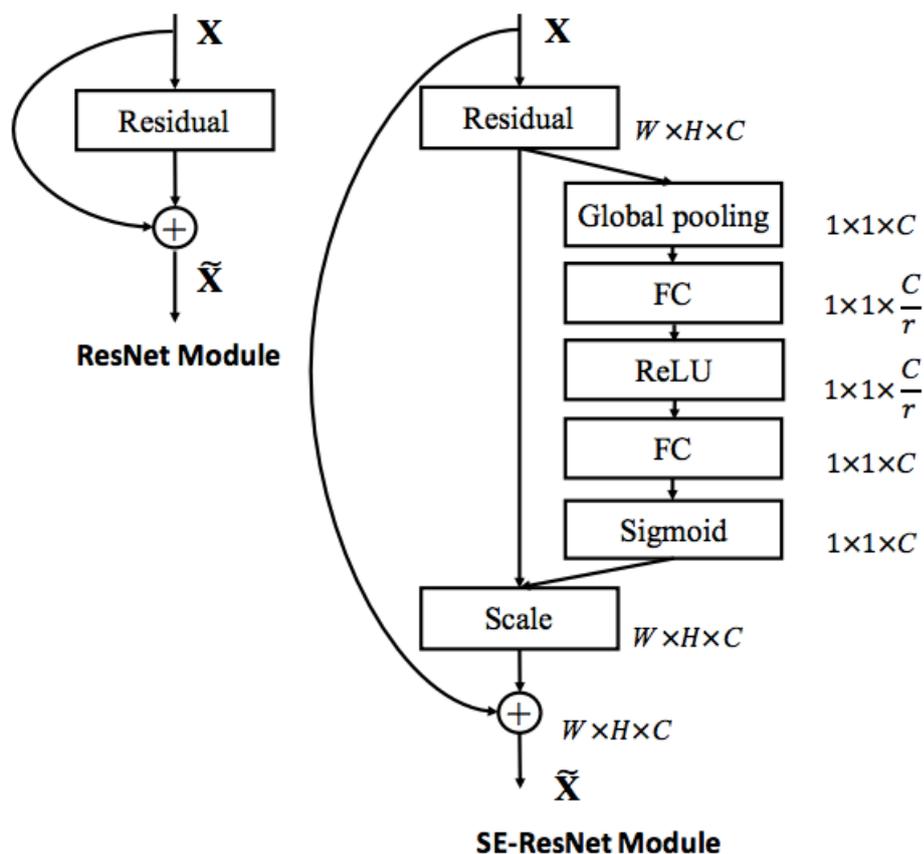


Рис. 2: Архитектура SENet

Красный — самый грубый уровень, который выполняет глобальное среднее объединение для каждой карты объектов, чтобы сгенерировать один выходной лоток.

Оранжевый — это второй уровень, который делит карту объектов на  $2 \times 2$  субрегиона, а затем выполняет среднее объединение для каждого субрегиона.

Синий — это третий уровень, который делит карту объектов на  $3 \times 3$  субрегиона, а затем выполняет среднее объединение для каждого субрегиона.

Зеленый — это лучший уровень, который делит карту объектов на  $6 \times 6$  субрегионов, а затем выполняет объединение для каждого субрегиона.

Затем выполняется свертка  $1 * 1$  для каждой объединенной карты признаков, чтобы уменьшить представление контекста до  $1/N$  от исходного (черного), если размер уровня пирамиды равен  $N$ . В этом при-

мере  $N = 4$ , потому что всего есть 4 уровня (красный, оранжевый, синий и зеленый). Если количество входных карт объектов равно 2048, то выходная карта объектов будет  $(1/4) \times 2048 = 512$ , т.е. 512 номеров выходных карт объектов.

Билинейная интерполяция выполняется для повышения дискретизации каждой карты объектов низкого размера, чтобы иметь тот же размер, что и исходная карта признаков (черная).

Все различные уровни карт признаков с повышенной дискретизацией объединяются с исходной картой признаков (черный цвет). Эти карты функций слиты как глобальный приоритет. Это конец "пирамидального" пулинг модуля в (с).

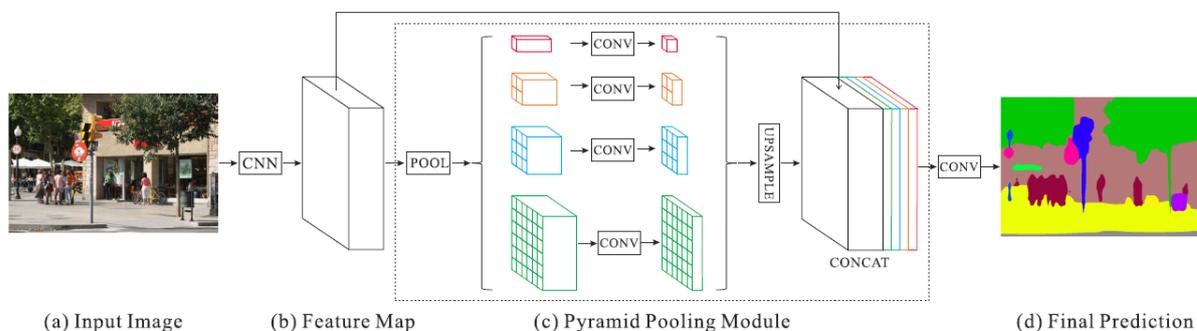


Рис. 3: Архитектура PSPNet

## 1.4. Rectified Adam оптимизация

Это вариация классического оптимизатора Adam, который обеспечивает автоматическую динамическую настройку адаптивной скорости обучения на основе их детального изучения влияния дисперсии и импульса во время обучения.

Авторы оригинальной статьи [6] обнаружили, что основная проблема заключается в том, что адаптивные оптимизаторы скорости обучения имеют слишком большое отклонение, особенно на ранних этапах обучения, и делают чрезмерные скачки, основываясь на ограниченных данных обучения и таким образом могут привести к плохим локальным оптимумам.

Следовательно, разогрев (warmup) (начальный период обучения с гораздо более низкой скоростью обучения) является требованием для адаптивных оптимизаторов, чтобы компенсировать чрезмерную дисперсию, когда оптимизатор работал только с ограниченным набором обучающих данных. Вот наглядное изображение того (рис. 4), что первоначально происходит с Adam оптимизатором без разогрева: в течение 10 итераций распределение градиента быстро возмущается:

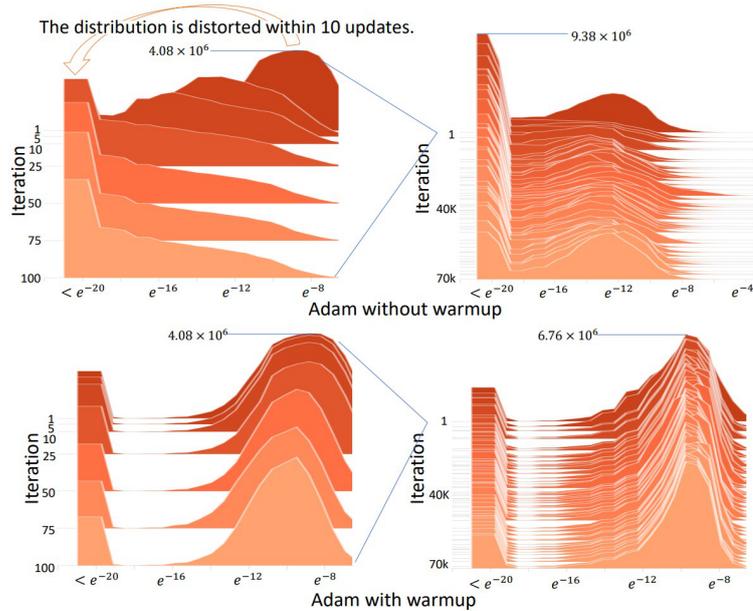


Рис. 4: Визуализация Adam оптимизатора с разогревом и без

Стандартный Adam и другие адаптивные оптимизаторы скорости обучения принимают неверные решения, основываясь на слишком малом количестве данных в начале обучения. Таким образом, без какой-либо разминки они, скорее всего, изначально попадут в плохой локальный оптимум, что сделает тренировочную кривую длиннее и сложнее из-за плохого старта.

Учитывая, что прогрев служит средством уменьшения дисперсии, но требуемая степень прогрева неизвестна и варьируется от набора данных к набору данных, авторы перешли к определению математического алгоритма, который будет служить в качестве динамического средства уменьшения дисперсии. Таким образом, они создали термин ”выпрямителя” (rectifier), который позволил адаптивному импульсу медленно, но

неуклонно работать до полного выражения в зависимости от базовой дисперсии.

## 1.5. Коэффициент Жаккара

В качестве функции потерь для одной из моделей в данном исследовании был использован коэффициент Жаккара разнообразия для двух выборок. Коэффициент измеряет сходство между наборами выборок и определяется как размер пересечения, деленный на размер объединения наборов выборок. Вычисляется по формуле 2:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \quad (2)$$

## 1.6. Псевдо-маркировка данных

Одна из основных проблем в области машинного обучения - отсутствие хорошей выборки данных для обучения моделей. Один из способов решения этой проблемы - псевдо-разметка данных [9].

В этом методе вместо ручной разметки неразмеченных данных мы даем приблизительные предсказания на основе размеченных данных. Пошаговый алгоритм описан на рисунке 5

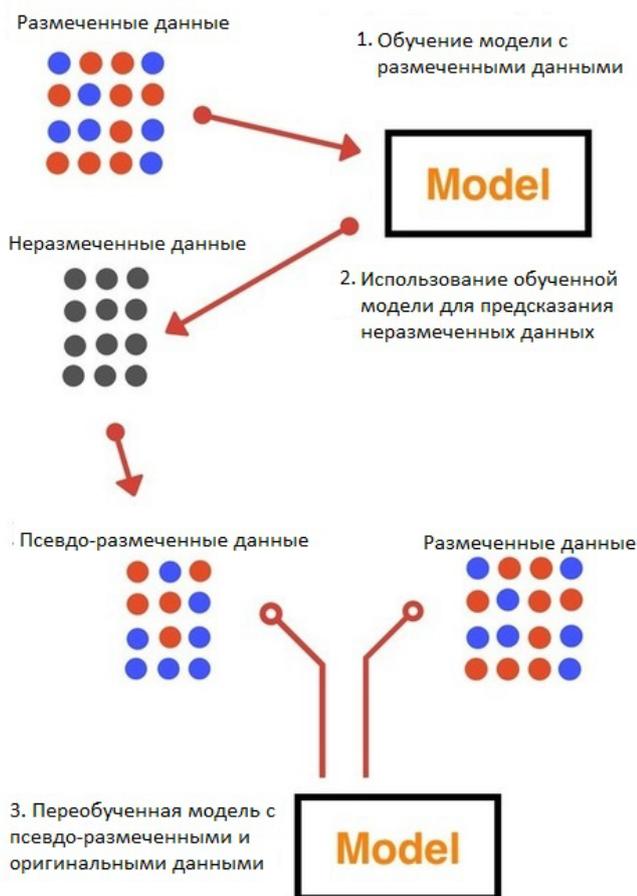


Рис. 5: Алгоритм псевдо-разметка данных

## 2. Данные

Для правильного подбора модели решения необходимо исследовать данные.

Как можно заметить, распределение количества изображений с дефектами отличается незначительно. Таким образом было предположено, что это хорошо сбалансированная проблема для задачи бинарной классификации(рис. 6).

Как мы видим, в исходных данных из обучающей выборки есть дисбаланс в распределении количества изображений по классам. Количество изображений с дефектом второго класса очень мало, а количество изображений с дефектом первого класса очень велико. Третий и четвертый классы несколько сбалансированы(рис. 7).

Помня о дисбалансе классификации по нескольким меткам, можно

	class_types	class_count
Defect_classes	Defect_class	6666.0
Non_defect_classes	Non_defect_class	5902.0

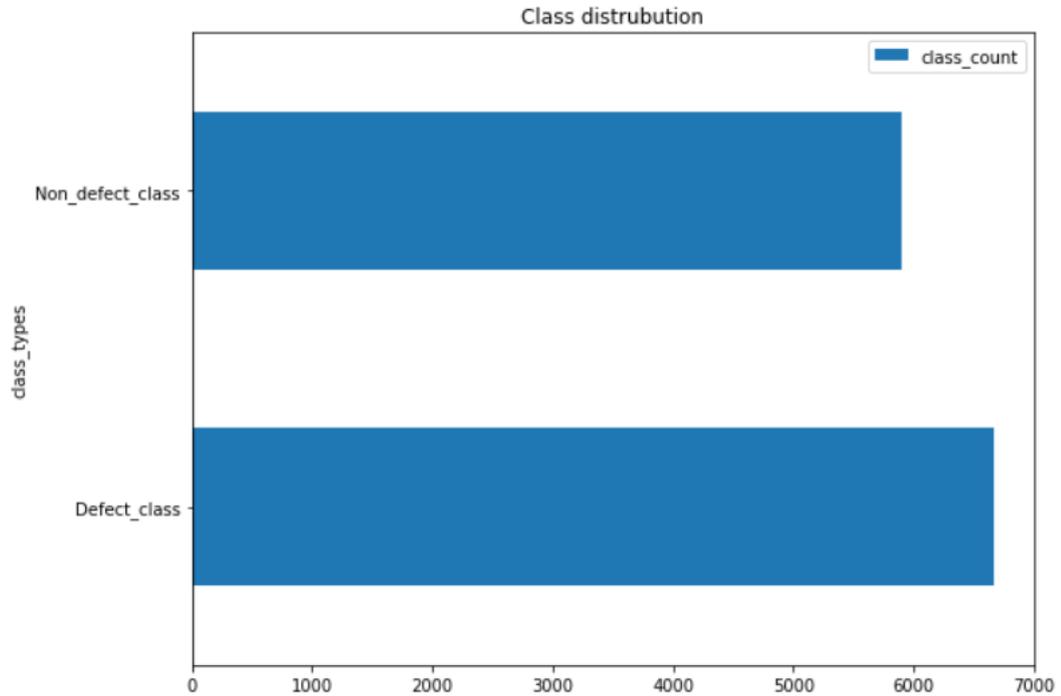


Рис. 6: Бинарная классификация изображений с дефектами и без

рассмотреть число дефектов каждого класса на изображении(рис. 8).

Давайте посмотрим на некоторые изображения каждого класса. Эту визуализацию можно легко сделать, маскируя заданные кодированные пиксели на изображениях данных из тренировочной выборки.

Оттенки красного цвета на изображениях описывают дефекты первого класса. Повреждения первого класса наименее выражены и почти аналогичны не дефектным изображениям(рис. 9).

Оттенки зеленого цвета на изображениях описывают дефекты второго класса. Мы можем заметить, что они похожи на дефекты первого класса, и между ними довольно сложно провести классификацию(рис. 10).

Пиксели, выделенные синим и желтым цветами на изображениях, описывают дефекты третьего класса. Можно заметить, что дефекты третьего класса гораздо обширнее и не похожи на первый и второй

	index	class_count
class_1	class1_Count	769
class_2	class2_Count	195
class_3	class3_Count	4759
class_4	class4_Count	516

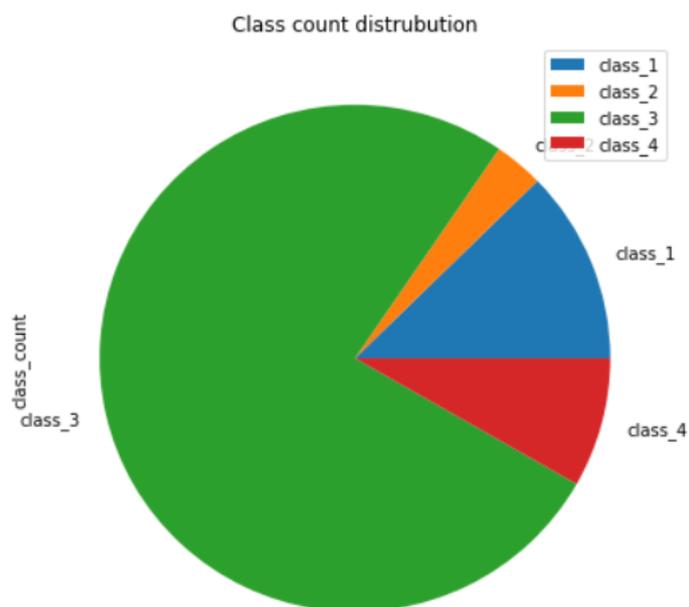


Рис. 7: Распределение изображений с дефектами по классам

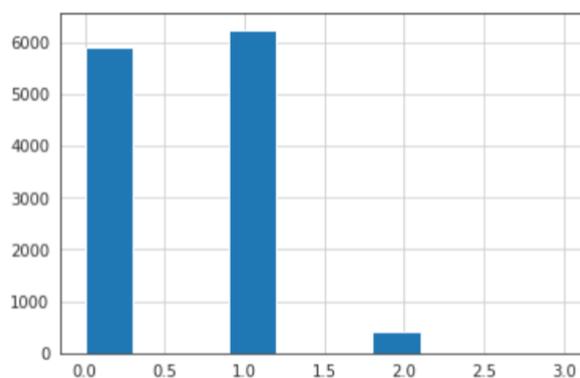


Рис. 8: Распределение количества дефектов по изображениям

классы(рис. 11).

Голубым цветом на изображениях выделены дефекты четвертого класса. Дефекты четвертого класса являются наиболее значительными.



Рис. 9: Дефекты первого класса



Рис. 10: Дефекты второго класса

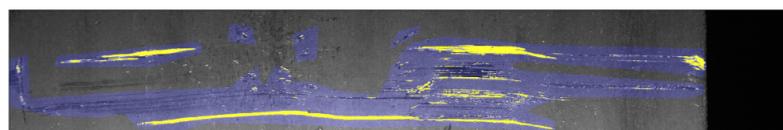


Рис. 11: Дефекты третьего класса

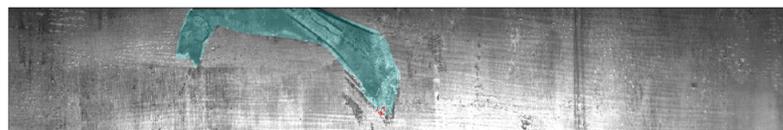
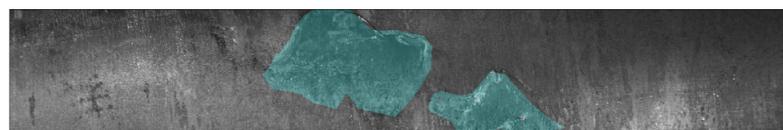


Рис. 12: Дефекты четвертого класса

Они могут быть легко классифицированы и сегментированы (рис. 12).

В итоге дефекты первого и второго классов похожи и менее дефект-

ны, в то время как дефекты третьего и четвертого классов менее похожи, но более значительны и следовательно их легко классифицировать.

### 3. Модель решения

Поскольку задача данного исследования связана с бинарной классификацией, многокомпонентной классификацией и регрессией, существует много подходов для ее решения. Ниже приведена часть из этих моделей, некоторые из них были использованы в исследовании.

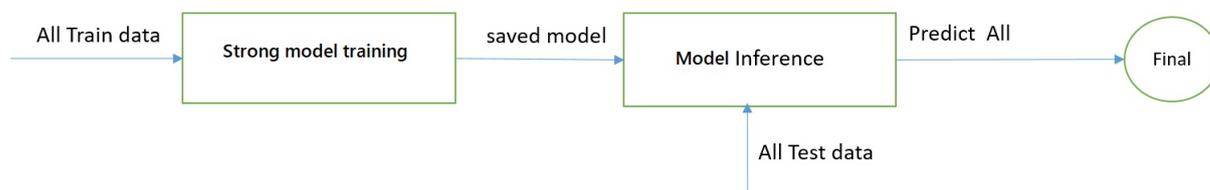


Рис. 13: Базовая модель с использованием сегментирующей компоненты

Первая и довольно эффективная стратегия - использование модели сегментации, которая разбивает изображение на области и помогает классифицировать объекты на нём.

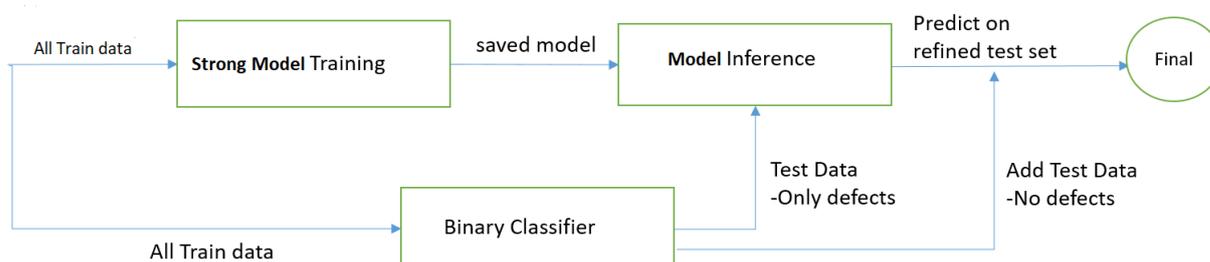


Рис. 14: Модель с использованием сегментирующей и бинарной компонент

Вторая стратегия имеет дело с построением модели сегментации наряду с бинарной классификацией. Модель сегментации (предварительно обученная) строится на всех обучающих данных. Аналогичным образом сохраняем бинарную модель классификации. Фильтруем изображения с помощью предварительно обученного двоичного классификатора и отправляем только дефектные изображения в модель сегментации. Это помогает быстрее тестировать результаты, а также фильтровать дефектные изображения из исходных данных (рис. 14).

Третья стратегия аналогична предыдущей, вместо фильтрации дефектных изображений используется предварительно обученная модель

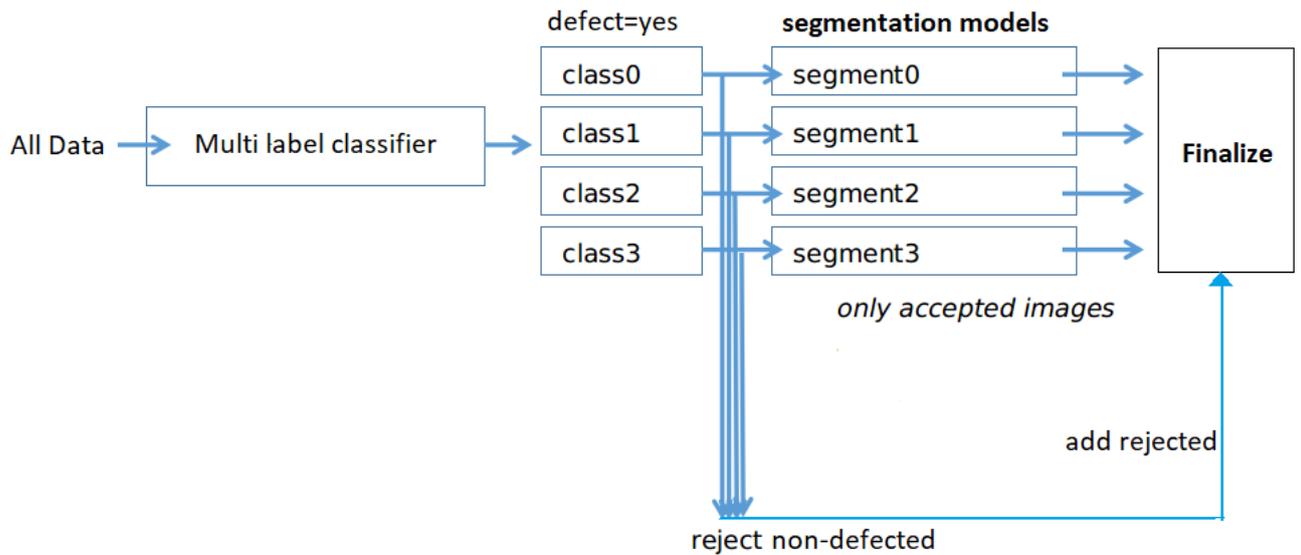


Рис. 15: Модель с использованием сегментирующей и многоклассовой компонент

классификации с несколькими метками для получения дефектных изображений. Так как это классификатор с несколькими метками, одно изображение может принадлежать более чем одному классу. Мы непосредственно берем выходные данные прохода классификации с несколькими метками для четырех моделей сегментации для каждого класса. Каждая модель представляет собой предварительно обученную модель сегментации с соответствующим классом дефектов(рис. 15).

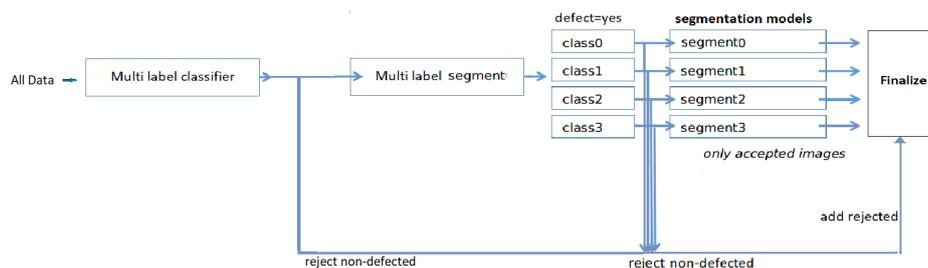


Рис. 16: Модель с использованием сегментирующей, бинарной и многоклассовой компонент

Четвертая модель является комбинацией двух предыдущих. Данный подход показал наилучший результат среди всех описанных стратегий. Процесс тестирования также занял меньше времени, поскольку были

отфильтрованы исправные изображения и отправлены только дефектные изображения в модель сегментации(рис. 16).

## 4. Исследование

### 4.1. Подготовка данных и инструментария для исследования

В качестве основы для исследования была использована четвертая модель. Исходные данные не являются данными временного ряда, поэтому можно случайным образом разбить их на тренировочную и тестовую выборки.

Изучая этот вопрос подробно были обнаружены существенные различия в тестовой и обучающей выборках, что учитывалось при обучении модели.

Обучение моделей осуществлялось с помощью сервиса Google Colab, который представляет собой специальный набор инструментов от компании Google, позволяющий запускать Python-код прямо в браузере. Данный сервис широко используется студентами, аналитиками данных и исследователями в области искусственного интеллекта. Он не требует какой-то дополнительной конфигурации среды исполнения, предоставляет свободный доступ к GPU и позволяет легко делиться написанным кодом.

Решение поставленной задачи осуществлено на языке Python с помощью фреймворка для глубокого обучения PyTorch. Дополнительно были использованы предобученные модели из Segmentation Models [10]. Полный исходный код решения можно найти в репозитории на GitHub (<https://github.com/avdeenkodmitry/steel-defect-detection>).

### 4.2. Многоклассовая классификация

В качестве классификатора использована нейронная сеть архитектуры UNet с энкодером SEnet154. Основной особенностью данной архитектуры является использование Squeeze-and-Excitation (SE) блоков. Существенным преимуществом этого подхода является возможность использования SE блоков с другими хорошо зарекомендовавшими себя моделями, такими как Inception и ResNet [8].

Обучение происходило с помощью k-fold cross-validation, разбивая выборку на 3 части, то есть  $k = 3$ . Увеличивая число k в разумных пределах, сильно растет вычислительная сложность модели, поэтому было выбрано оптимальное значение в данной ситуации.

Применена техника изменения исходного размера изображения до  $128 * 800$  пикселей. Это позволило ускорить качество обучения, оптимизировав потребление памяти у видеокарты, использовавшейся для обучения.

Также была применена техника аугментации данных для увеличения объема обучающей выборки.

В качестве оптимизатора модели использован RAdam (Rectified Adam). Это новый вариант классического оптимизатора Adam, который обеспечивает автоматическую динамическую настройку адаптивной скорости обучения на основе их детального изучения влияния дисперсии и импульса во время обучения.

Применена BCE (Binary Cross-Entropy) функция потерь для данной модели. Эта функция независима от каждого векторного компонента, что позволяет использовать ее для классификации по нескольким классам. Техника ТТА (test-time augmentation) для данной модели использована не была.

Все мета-параметры модели перечислены в таблице 1

Модель	SENet154
Размер изображения	Сжатие до $128 * 800$ пикселей
Аугментация	Нормализация
Оптимизатор	RAdam
Функция потерь	BCE
Аугментация во время теста	Не использовалась

Таблица 1: Параметры для модели многоклассовой классификации

Для обучения модели была использована реализация сети UNet с энкодером SENet154 из библиотеки Segmentation models. SENet154 предобучен на базе изображений ImageNet и содержит 113 миллионов параметров. В качестве параметров learning rate (коэффициент скорости

обучения) и weight decay (сокращение весов) оптимизатора RAdam использованы следующие значения: для энкодера установлены

$$\text{learning rate} = 7e - 5, \text{weight decay} = 3e - 5,$$

для декодера

$$\text{learning rate} = 3e - 3, \text{weight decay} = 3e - 4.$$

Обучение проводилось на 30 эпохах с оценкой показателей loss (функции потерь), accuracy (точность) и precision (полнота). При малом улучшении параметров обучающей выборки или при ухудшении качества параметров тестовой выборки процесс обучения модели прерывался во избежание переобучения. На 21-ой эпохе была остановлена тренировка и оценена функция потерь (Рис. 17). На 4.2 представлены две последние эпохи обучения.

Epoch 20/30

```
640/640 [=====] - 639s 234ms/step - loss:
0.1597 - acc: 0.9453 - recall: 0.9512 - val_loss: 0.1723 - val_acc:
0.9321 - val_recall: 0.9173
```

Epoch 21/30

```
640/640 [=====] - 634s 624ms/step - loss:
0.1397 - acc: 0.9446 - recall: 0.9517 - val_loss: 0.1384 - val_acc:
0.9021 - val_recall: 0.9612
```

### 4.3. Многоклассовая сегментация

Для многоклассовой сегментации использован ансамбль из 3-х моделей.

1. PSPNet с SE-ResNext101-32x4d энкодером
2. FPN с SENet154 энкодером
3. PSPNet с SENet154 энкодером

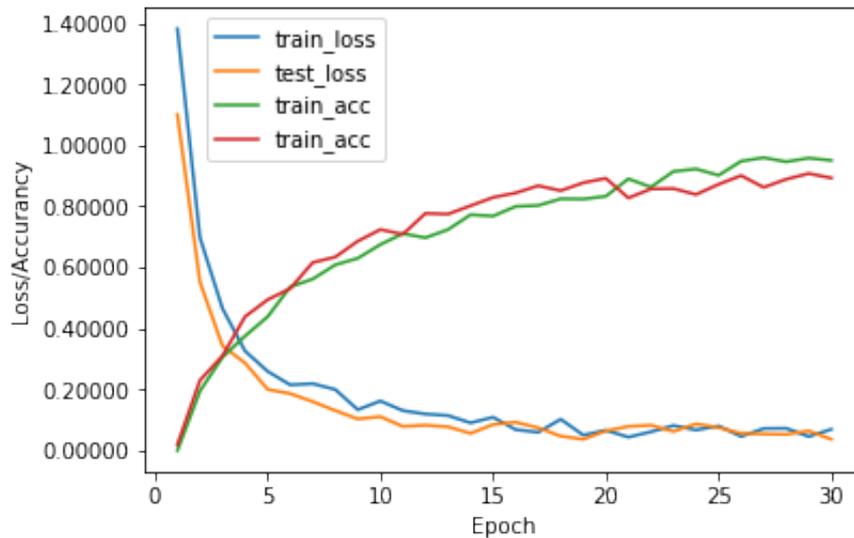


Рис. 17: Изменение функции потерь и точности обучения от эпохи к эпохе

Первая модель обучена на полных изображениях с функциями потерь ВСЕ и Jaccard. Вторая и третья модели обучены на обрезанных изображениях размера  $256 * 256$  с функцией потерь ВСЕ.

Помимо стандартных методов аугментации данных был применен еще один метод, реализованный в рамках этой задачи. Обрезая каждое изображение из исходной выборки до размера  $256 * 256$ , но при этом принимая эту картинку в тренировочную выборку только в том случае, если дефект попал на изображение. Это сделано для того, чтобы модель обучалась только на изображениях, содержащих маску дефекта.

Также был применен еще один способ для улучшения качества модели и увеличения скорости обучения. В качестве основы для двух последних моделей из ансамбля были использованы предобученный классификатор Senet154 из пункта 3.2.

В качестве дополнительной оптимизации были удалены небольшие дефекты и пустоты на детектированных объектах размером меньше, чем 128 пикселей. Это позволило сильно улучшить качество модели и убрать ложноположительные предсказания.

Нужно отметить, что на самом деле данный этап фактически используется для того, чтобы отбросить изображения без дефектов и передать в модель бинарной сегментации только дефектные изображения.

Все мета-параметры модели перечислены в таблице 2

Модели	PSPNet с SE-ResNext101-32x4d FPN с SENet154 PSPNet с SENet154
Размер изображения	Полные изображения Сегменты 256 * 256 пикселей
Аугментация	Модифицированная аугментация
Оптимизатор	Adam, RAdam
Функция потерь	BCE, BCE + Jaccard
Аугментация во время теста	Горизонтальный поворот

Таблица 2: Параметры для модели многоклассовой сегментации

По аналогии с предыдущим пунктом были использованы предобученные модели из библиотеки Segmentation models и аналогичные значения для параметров оптимизатора. Обучение проводилось на 30 эпохах для каждой модели с оценкой функции потерь, точности и полноты. При малом улучшении параметров на обучающей выборке или при ухудшении качества процесс обучения модели прерывался во избежании переобучения. Для PSPNet модели с SE-ResNext101-32x4d энкодером была остановлена тренировка и оценена функция потерь (Рис. 18) на 23-ой эпохе, для FPN модели с SENet154 энкодером - на 20-ой эпохе, для PSPNet модели с SENet154 энкодером - на 27-ой эпохе.

#### 4.4. Бинарная сегментация

На третьем этапе были использованы бинарные модели сегментации, обученные для каждого класса отдельно. Для первого класса была использована нейронная сеть архитектуры UNet с SE-ResNext50 энкодером, для второго и третьего FPN с SE-ResNext50 энкодером и для четвертого класса UNet с SeNet154 энкодером.

Для каждого класса изображения была использована отдельная модель сегментации. Причем модель каждого класса обучалась только на изображениях, содержащих дефекты этого класса. Обучение происходило на обрезанных изображениях размером 256 \* 768 с применением

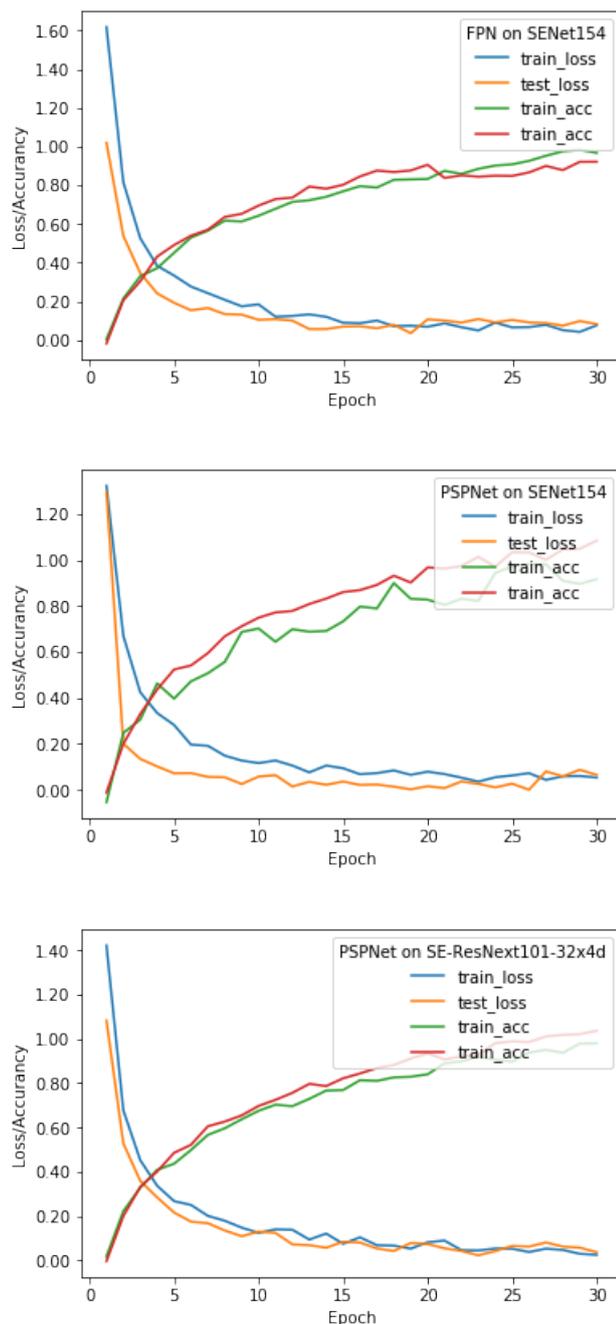


Рис. 18: Изменение функции потерь и точности обучения от эпохи к эпохе для многоклассовой сегментации

техники включения только дефектных участков, описанной в 3.3.

В качестве оптимизатора модели был использован RAdam. Функция потерь:  $(1-Dice)0.4 + 0.6VCE$ . Также применены различные техники аугментации данных, такие как горизонтальные и вертикальные перевороты, увеличение яркости и сжатие изображений.

По аналогии с предыдущими пунктами использованы предобучен-

ные модели из пакета Segmentation models. Аналогичные значения использованы для параметров оптимизатора. Обучение проводилось на 25 эпохах для каждой модели с оценкой функции потерь и коэффициента Сёренсена. При малом улучшении параметров на обучающей выборке или при ухудшении качества параметров тестовой выборки тренировочный процесс прерывался во избежание переобучения. Для модели сегментации дефектов первого класса UNet с SE-ResNext50 энкодером была остановлена тренировка и оценена функции потерь на 17-ой эпохе, для моделей сегментации FPN с SE-ResNext50 энкодером второго и третьего классов - на 19-ой и 21-ой эпохах соответственно, для PSPNet модели с SENet154 энкодером четвертого класса- на 24-ой эпохе (Рис. 19, 20, 21, 22).

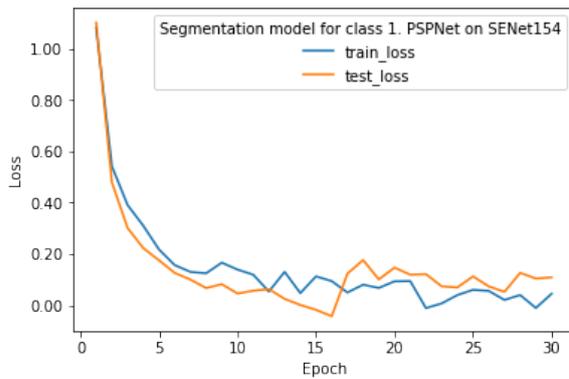


Рис. 19: Бинарная сегментация для дефектов первого класса

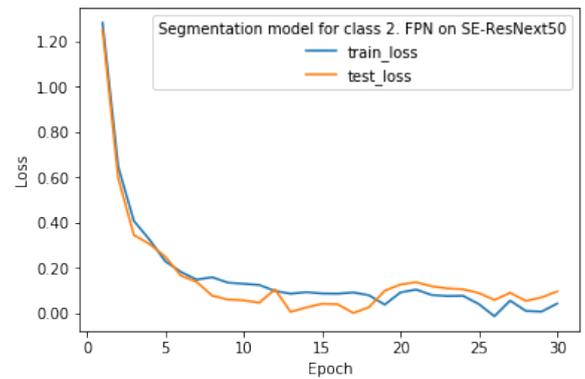


Рис. 20: Бинарная сегментация для дефектов второго класса

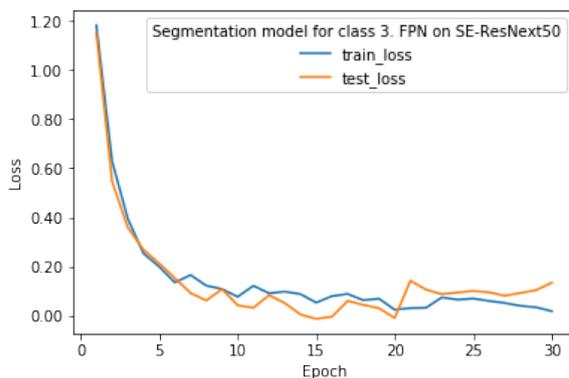


Рис. 21: Бинарная сегментация для дефектов третьего класса

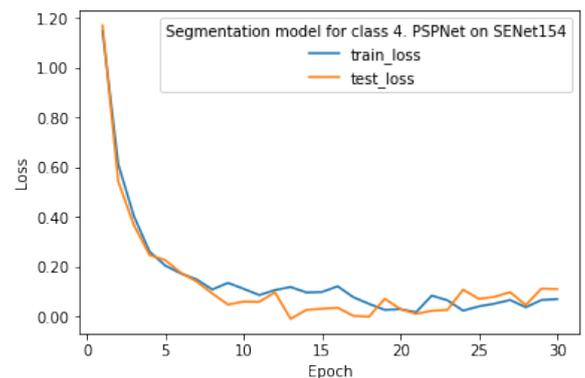


Рис. 22: Бинарная сегментация для дефектов четвертого класса

## 4.5. Результаты

На рисунке ниже изображено изменение значения коэффициента Сёрнсена в полученной модели от дополнительных изменений, добавляемых постепенно в модель.

1. Многоклассовая классификация на SENet154 + многоклассовая сегментация
2. Добавлены псевдоклассы для классификатора
3. Добавлены две дополнительные модели для многоклассовой сегментации.
4. Применена оптимизация с пропуском детектируемых объектов размером меньше 128 пикселей.
5. Добавлена бинарная сегментация на последнем этапе
6. Бинарная классификация улучшена с помощью ансамбля моделей на последнем этапе

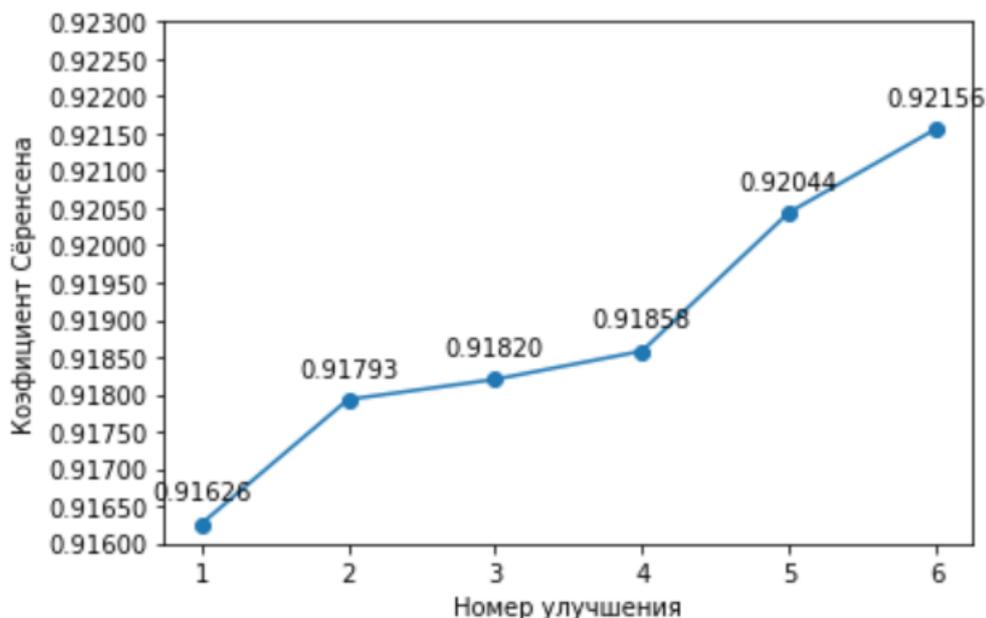


Рис. 23: График значений коэффициента Сёрнсена

## Выводы

В ходе проведенного исследования сделан вывод о том, что модель, описанная в Главе 2 хорошо подходит для решения задачи детектирования дефектов на изображениях листов стали. Для получения наилучшего результата необходимо использовать различные компоненты, такие как многоклассовая классификация, бинарная и многоклассовая сегментации.

## Заключение

Целью данной выпускной квалификационной работы являлись исследование задачи обнаружения дефектов на листах стали, а также практическая реализация модели решения на языке Python с использованием фреймворков и библиотек глубокого обучения.

На первом этапе были формально определены задача и исходные данные для дальнейшего исследования. Произведено подробное описание изображений, представленных в качестве обучающей и тестовой выборки. Исследованы структура выборки, распределение изображений по классам, отличия классов дефектов.

Представлено несколько моделей, которые были использованы для решения данной задачи, описаны их недостатки и особенности применения, а также выбрана модель для практической реализации.

На следующем этапе была программно реализована модель и подробно описаны алгоритмы для каждой части полученного решения.

Заключительным этапом стали подведение итогов и оценка качества предсказаний построенной модели.

В результате данной выпускной квалификационной работы изучены различные методы машинного обучения для анализа изображений, области их применения и способы решения задач данными методами. Подробно проанализирована и описана задача обнаружения дефектов на листах стали от компании Северсталь. В практической части выпускной квалификационной работы предложено несколько подходов для решения поставленной задачи, реализовано несколько моделей на языке Python.

## Список литературы

- [1] Aleatoric uncertainty estimation with test-time augmentation for medical image segmentation with convolutional neural networks / Guotai Wang, Wenqi Li, Michael Aertsen et al. // arXiv e-prints. — 2018. — jul. — P. arXiv:1807.07356. — 1807.07356.
- [2] Baez John C., Dolan James. Higher-dimensional algebra and topological quantum field theory // Journal of Mathematical Physics. — 1995. — nov. — Vol. 36, no. 11. — P. 6073–6105. — q-alg/9503002.
- [3] Goodfellow Ian, Bengio Yoshua, Courville Aaron. Deep Learning. — MIT Press, 2016. — <http://www.deeplearningbook.org>.
- [4] Kingma Diederik, Ba Jimmy. Adam: A Method for Stochastic Optimization // International Conference on Learning Representations. — 2014. — 12.
- [5] Lab Stanford Vision. Home page // ImageNet. — 2020. — URL: <http://www.image-net.org/> (online; accessed: 02.05.2020).
- [6] On the Variance of the Adaptive Learning Rate and Beyond / Liyuan Liu, Haoming Jiang, Pengcheng He et al. // arXiv e-prints. — 2019. — aug. — P. arXiv:1908.03265. — 1908.03265.
- [7] Ronneberger Olaf, Fischer Philipp, Brox Thomas. U-Net: Convolutional Networks for Biomedical Image Segmentation // arXiv e-prints. — 2015. — may. — P. arXiv:1505.04597. — 1505.04597.
- [8] Squeeze-and-Excitation Networks / Jie Hu, Li Shen, Samuel Albanie et al. // arXiv e-prints. — 2017. — sep. — P. arXiv:1709.01507. — 1709.01507.
- [9] Wilson Garrett, Cook Diane J. Multi-Purposing Domain Adaptation Discriminators for Pseudo Labeling Confidence // arXiv e-prints. — 2019. — jul. — P. arXiv:1907.07802. — 1907.07802.

- [10] Yakubovskiy Pavel. Segmentation Models Pytorch. — [https://github.com/qubvel/segmentation\\_models.pytorch](https://github.com/qubvel/segmentation_models.pytorch). — 2020.