МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»**

Кафедра математического анализа

«Триангуляция областей»

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 421 группы

направление 02.03.01 — Математика и компьютерные науки

механико-математического факультета

Праслова Ильи Викторовича

Научный руководитель
доцент, к.ф.-м.н. Матвеева Ю. В.

Заведующий кафедрой
зав. кафедрой, к.ф.-м.н., доцент Разумовская Е. В.

Саратов 2026

Введение. Современные методы компьютерного зрения активно используют нейронные сети для детектирования ключевых точек, позволяющих восстанавливать структуру объектов, анализировать форму и движение, а также проводить 3D-реконструкцию. Архитектура U-Net особенно эффективна благодаря сочетанию глубокого извлечения признаков и точного восстановления пространственной структуры, что критично для задач, требующих пиксельной точности. Построение триангуляции Делоне по предсказанным точкам обеспечивает геометрически устойчивое разбиение области, удобное для анализа формы, генерации сеток и визуализации. Автоматическое определение ключевых точек и их триангуляция востребованы в следующих областях: компьютерное зрение, медицинская визуализацияй, робототехника и навигация.

Несмотря на значительные успехи в области компьютерного зрения, актуальной остаётся задача точного и устойчивого определения ключевых точек на изображениях с последующей геометрической интерпретацией результатов. Для этого требуется не только корректная архитектура нейронной сети, но и оптимальная функция потерь, методы обучения и процедуры оценки качества полученной триангуляции. Целью работы является разработка и обучение сверточной нейронной сети типа U-Net с помощью языка программирования Python и фреймворка PyTorch для предсказания ключевых точек изображения, на основе которых строится триангуляция.

Основное содержание работы. Работа состоит из шести глав.

Первая глава работы посвящена триангуляции и связанным понятиям. В ней вводятся базовые геометрические определения: узел как точка схождения рёбер, ребро как отрезок, соединяющий вершины, треугольник как элементарная ячейка, ограниченная тремя рёбрами, и триангуляция как разбиение заданной области на непересекающиеся треугольники, объединение которых точно воспроизводит исходную область. Центральное место занимает триангуляция Делоне.

Определение (Триангуляция Делоне). Пусть P — множество точек на плоскости. Триангуляция Делоне — это такая триангуляция множества P, что для любого треугольника окружность, описанная около него, не содержит других точек множества P.

Важнейшим следствием условия пустой окружности является свойство максимизации минимального угла. Это исключает появление чрезмерно вытянутых ячеек, что критически важно для численной устойчивости и точности последующих расчётов.

Теорема (Оптимальность Делоне-триангуляции по сумме минимальных углов). Пусть P — конечное множество точек в общем положении (никакие четыре не лежат на одной окружности). Обозначим через D её Делоне-триангуляцию, а через T любую другую триангуляцию множества P. Тогда

1. Сумма всех минимальных углов в D не меньше, чем сумма минимальных углов в любой другой триангуляции T:

$$\sum_{\triangle \in D} \min \angle(\triangle) \;\geq\; \sum_{\triangle \in T} \min \angle(\triangle).$$

2. Если в триангуляции T есть хотя бы одно «нелегальное» ребро (то есть нарушающее критерий Делоне), то неравенство строгое:

$$\sum_{\triangle \in D} \min \angle(\triangle) \;>\; \sum_{\triangle \in T} \min \angle(\triangle).$$

Представленные теоретические результаты формируют строгий математический аппарат, необходимый для последующей разработки алгоритмов автоматической локализации ключевых точек и их применения в задачах компьютерного зрения, трёхмерной реконструкции и геометрического моделирования.

Вторая глава содержит теоретические основы представления изображений и понятие ключевых точек.

Цифровое изображение можно рассматривать как двумерную дискретную функцию

$$I(x, y),$$

где каждому набору координат (x, y) соответствует значение яркости или цветовой вектор. Наиболее распространённые способы представления изображений — это изображения в градациях серого и цветные изображения, описываемые несколькими компонентами.

Определение (Цветовая модель RGB). Одной из наиболее распространённых цветовых моделей является модель RGB (Red, Green, Blue). В рамках этой модели каждый пиксель представляется трёхмерным вектором

$$I(x, y) = (R, G, B),$$

где R, G и B — интенсивности красной, зелёной и синей компонент соответственно. Модель RGB основана на аддитивном принципе смешения цветов: увеличение интенсивностей отдельных каналов приводит к более яркому результирующему цвету, а максимальные значения всех трёх компонент соответствуют белому цвету.

Каждая компонентная матрица рассматривается независимо, что удобно для выполнения операций фильтрации, выделения признаков и вычисления градиентов. Например, при поиске границ можно анализировать каждый канал отдельно, а затем объединять результаты.

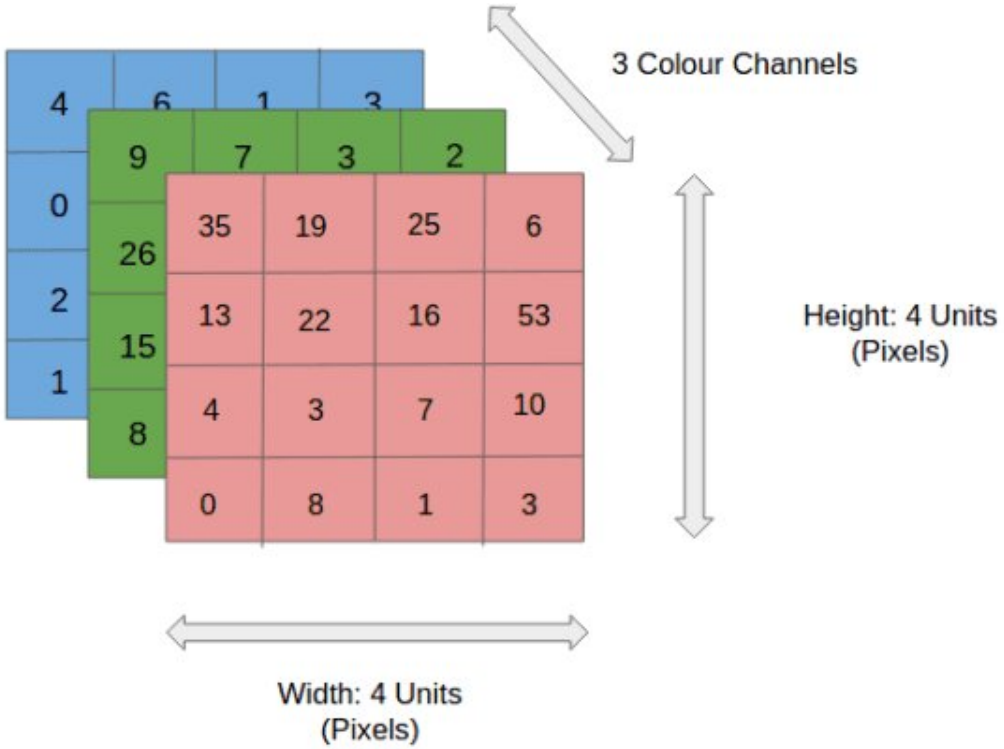

Рисунок 1 — RGB-компоненты изображения

Также рассматриваются базовые операции обработки изображений, такие как изменение разрешения изображения, выделение границ и поиск ключевых точек.

Определение (Ключевые точки изображения). Ключевые точки представляют собой локальные особенности изображения, обладающие высокой информативностью и устойчивостью к различным преобразованиям. Формально ключевая точка $p = (x, y)$ определяется в локальной окрестности $N(p)$ как такая точка, для которой некоторая функция интереса $F(I, p)$ достигает экстремального значения:

$$F(I, p) = \max/\min_{q \in N(p)} f(I, q),$$

где I — матрица интенсивностей изображения, а $f(I, q)$ — функция, оценивающая локальные изменения яркости или текстуры.

Математические основы обучения искусственных нейронных сетей описаны в третьей главе. Вводятся важнейшие определения, такие как искусственный нейрон, функция активации, функция потерь.

Определение (Функция потерь). Функцией потерь называется отображение

$$\mathcal{L}(y, \hat{y}) : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_{\geq 0},$$

которое количественно измеряет расхождение между истинным значением целевой переменной y и предсказанием модели $\hat{y}$. Чем меньше значение функции потерь, тем лучше модель аппроксимирует наблюдаемые данные. Функция потерь служит критерем качества модели и оптимизируется в процессе обучения.

Из метода максимального правдоподобия выводятся функции потерь, например, отрицательное лог-правдоподобие:

$$\mathcal{L}(\theta) = -\sum_{i=1}^{N} \log p_\theta(y_i \mid x_i).$$

Во многих задачах эта величина совпадает с классическими функциями потерь. Например, в бинарной классификации отрицательное лог-правдоподобие приводит к кросс-энтропии

$$\mathcal{L}(y, \hat{y}) = -\left[y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})\right].$$

В регрессионных задачах обычно используется среднеквадратичная ошибка

$$\mathcal{L}(y, \hat{y}) = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2.$$

Задача обучения нейронной сети сводится к минимизации функции потерь по параметрам модели. Поскольку аналитическое решение такой задачи для многослойных нелинейных моделей практически недостижимо, оптимизация проводится численно — с помощью итеративных методов первого порядка. Базовым подходом является градиентный спуск.

На каждой итерации параметры θ обновляются в направлении, противоположном градиенту функции потерь:

$$\theta^{(t+1)} = \theta^{(t)} - \eta \, \nabla_\theta \mathcal{L}(\theta^{(t)}),$$

где $\eta > 0$ — коэффициент скорости обучения, определяющий величину шага.

Для выполнения описанных обновлений необходимо вычислять градиенты функции потерь по всем параметрам сети. Это осуществляется методом обратного распространения ошибки (backpropagation), основанным на последовательном применении правила цепочки.

Рассмотрим линейный слой вида

$$z = Wx + b, \qquad y = \sigma(z),$$

где σ — нелинейная функция активации. Обозначим

$$\delta = \frac{\partial \mathcal{L}}{\partial z}$$

— локальную ошибку данного слоя. Тогда градиенты параметров и входа слоя выражаются следующим образом:

$$\frac{\partial \mathcal{L}}{\partial W} = \delta x^\top, \qquad \frac{\partial \mathcal{L}}{\partial x} = W^\top \delta.$$

Тем самым ошибка распространяется от выходного слоя к входному, что позволяет эффективно вычислить градиенты для всех матриц весов и смещений в сети.

Четвертая глава вводит базовые определения, связанные со сверточными нейронными сетями - специальными моделями, широко используемыми для работы с изображениями. Рассматривается архитектура нейронной сети U-Net и принципы её функционирования.

Архитектура U-Net имеет характерную U-образную форму, что отражено в её названии. Она состоит из двух симметричных частей: сужающая часть (последовательно уменьшает пространственное разрешение, извлекает семантические признаки высокого уровня), расширяющая часть (увеличивает разрешение и восстанавливает детальные признаки объекта).

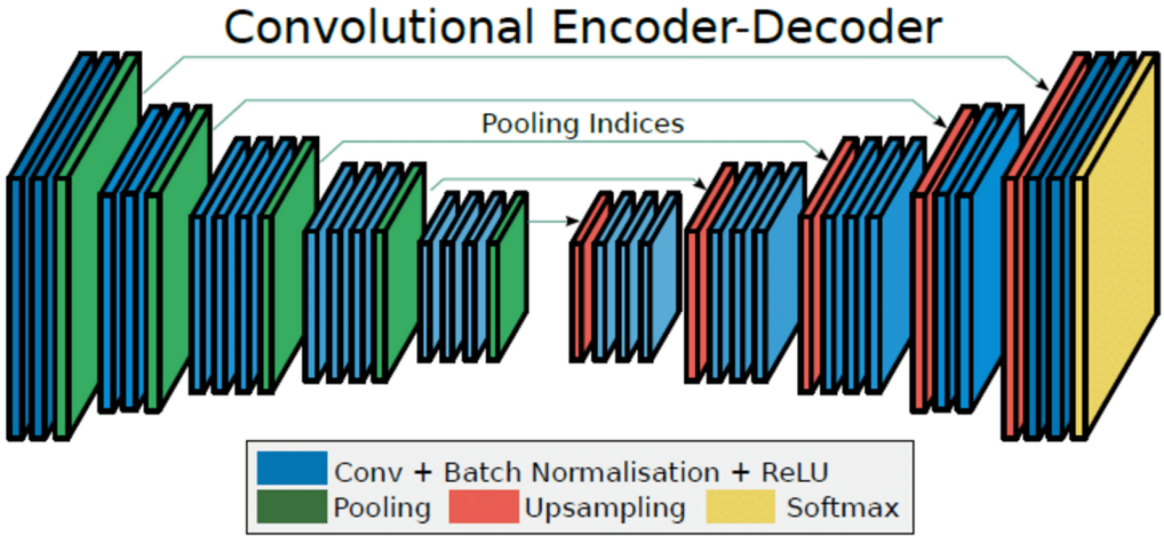

Рисунок 2 — Архитектура нейросети U-Net

Практическая часть работы начинается с пятой главы. В ней изложен процесс подготовки данных, для обучения модели.

В ходе экспериментов использовалась подвыборка набора изображений ImageNet, содержащая цветные RGB-изображения.

Данные были предварительно разделены на три части: обучающую, валидационную и тестовую. Разбиение выполнялось в пропорции 16 000/3 000/3 000 изображений соответственно. Все изображения были приведены к единому размеру 512×512 пикселей. На подготовленном датасете осуществлялось обучение нейросетевой архитектуры типа **U-Net**, выбран-

ной благодаря её эффективности в задачах пиксельного предсказания и сегментации. Для формирования набора разметки была разработана процедура автоматического выделения ключевых точек на изображениях.

Для каждого изображения выполняются следующие действия: загружается цветное изображение с проверкой соответствия требуемому разрешению и связанный файл с ключевыми точками, содержащий их координаты и параметр важности, каждая точка преобразуется в гауссово пятно, амплитуда которого пропорциональна важности, формируя тепловую карту того же размера, что и исходное изображение, интенсивность тепловой карты и пиксели изображения нормализуются в диапазон [0, 1], после чего применяются необходимые преобразования для подачи в модель, включая транспонирование каналов. В результате каждая выборка представляется в виде пары тензоров PyTorch (изображение, тепловая карта), полностью готовых к обучению нейронной сети.

Архитектура нейронной сети конкретной технической реализации проводится в шестой главе, после чего проводится обучение сети, построение триангуляции по предсказанным точкам и анализ качества полученных результатов.

Модель `UNetSmall` отличается от стандартной U-Net компактностью и упрощением архитектуры. Базовое число каналов задаётся параметром `base=32`, что делает модель легче по сравнению с классической U-Net, где обычно используют 64 и более фильтров на первом уровне.

Обучение модели `UNetSmall` выполняется с использованием пользовательского датасета `KeypointHeatmapDataset`, содержащего изображения и соответствующие им тепловые карты ключевых точек. Датасет разделён на обучающую и валидационную выборки, а загрузка данных организована через `DataLoader` с батчами заданного размера и параллельной обработкой.

В качестве функции потерь используется комбинация взвешенной бинарной кросс-энтропии (`BCEWithLogitsLoss`) и среднеквадратичной ошибки (`MSE`). Взвешивание BCE выполняется с учётом значений тепловой карты, что позволяет уделять больше внимания более значимым ключевым точкам. Итоговая функция потерь вычисляется как

$$\mathcal{L} = \mathcal{L}_{\text{BCE}} + \mathcal{L}_{\text{MSE}}$$

где первое слагаемое — взвешенная бинарная кросс-энтропия:

$$\mathcal{L}_{\text{BCE}} = \frac{1}{N} \sum_{i=1}^{N} w_i \cdot \Big[- y_i \log \hat{y}_i - (1 - y_i) \log(1 - \hat{y}_i) \Big],$$

N — число пикселей, y_i — значение эталонной тепловой карты в пикселе i, $\hat{y}_i = \sigma(\text{логит}_i)$ — предсказанное значение после сигмоиды, а $w_i = 1 + \alpha \cdot y_i$ — пиксельный вес, усиливающий значение важных точек.

Второе слагаемое — среднеквадратичная ошибка между предсказанным и эталонным распределением:

$$\mathcal{L}_{\text{MSE}} = \frac{\lambda_{\text{mse}}}{N} \sum_{i=1}^{N} (\hat{y}_i - y_i)^2,$$

где λ_{mse} — коэффициент влияния MSE.

`BCE` хорошо справляется с классификацией пикселей как «ключевая точка / не ключевая точка», но менее чувствителен к точным значениям предсказанной вероятности. `MSE` измеряет разницу между фактическими значениями тепловой карты и предсказанными, усиливая обучение на точности амплитуды гауссианов. `BCE` в комбинации с `MSE` помогает сети не только правильно локализовать ключевые точки, но и формировать корректную форму и высоту гауссианов, что улучшает визуализацию и последующую интерпретацию предсказаний.

Обученная таким образом сеть предсказывает набор ключевых точек для изображения. После этого применяется алгоритм Делоне для построения триангуляции. На основе полученных треугольников формируется мозаика: для каждого треугольника создаётся маска, вычисляется средний цвет исходного изображения внутри треугольника и треугольник закрашивается этим цветом на новом изображении.

В результате получается визуализация исходного изображения в виде мозаики, где каждая область триангуляции окрашена усреднённым цветом. Та-

кой подход наглядно демонстрирует распределение ключевых точек и позволяет визуализировать сегментацию изображения через триангуляцию.

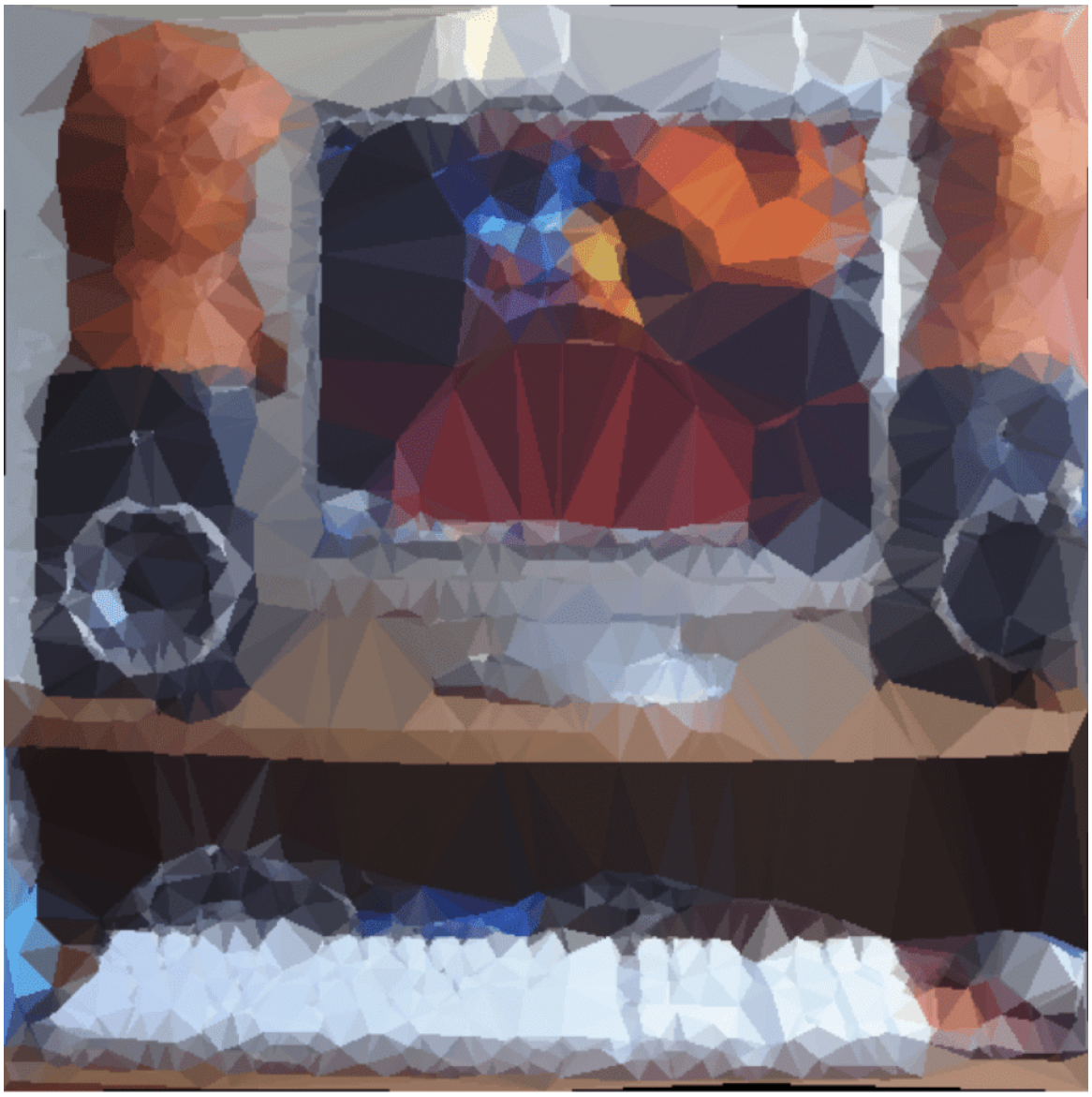

Рисунок 3 — Обработанное изображение

Заключение. В данной работе исследовано применение сверточных нейронных сетей для предсказания ключевых точек изображения и последующего построения по ним триангуляции. Основной акцент сделан на архитектуре U-Net, которая успешно сочетает глубокое извлечение признаков с точным восстановлением пространственной структуры, что делает её оптимальным решением для задач пиксельной локализации.

Изучены принципы представления изображений в виде многоканальных матриц, особенности их обработки сверточными сетями и методы стабилиза-

ции обучения через нормализацию и функции активации. Подготовленный набор данных состоял из пар изображение-ключевые точки, которые преобразовывались в гауссовы тепловые карты. Такой формат представления обеспечил стабильное распространение градиентов и повысил точность определения позиций характерных точек.

Экспериментальная часть реализована на языке Python с использованием фреймворка PyTorch. Это позволило гибко обрабатывать данные, эффективно обучать архитектуру U-Net, управлять процессом оптимизации и задействовать графические ускорители. В качестве функции потерь применялась бинарная кросс-энтропия в сочетании с градиентным оптимизатором. Мониторинг динамики потерь подтвердил стабильное снижение ошибки и корректность процесса обучения.

На основе предсказанных ключевых точек выполнена триангуляция методом Делоне. Качество работы системы оценивалось по метрикам точности локализации, включая среднюю евклидову ошибку и долю корректно определённых точек, а также по геометрическим характеристикам полученной сетки, таким как регулярность треугольников и равномерность длин рёбер. Эксперименты показали, что предложенный подход обеспечивает устойчивые результаты и достоверно восстанавливает структурные элементы изображений.

Разработанная система демонстрирует успешную интеграцию нейросетевых методов локализации с классическими геометрическими алгоритмами, обеспечивая высокую гибкость реализации, воспроизводимость экспериментов и масштабируемость вычислений. В дальнейшем планируется исследовать более совершенные архитектуры, внедрить методы автоматической генерации и аугментации данных, адаптировать подход для трёхмерной реконструкции и провести углублённый анализ метрических и топологических свойств получаемых триангуляций.

Полученные результаты подтверждают эффективность сочетания глубокого обучения и вычислительной геометрии при анализе визуальных данных. Разработанный подход создаёт прочную основу для дальнейшего применения в задачах компьютерного зрения, медицинской диагностики, робототехники и моделирования сложных поверхностей.