УДК 004.932

# ПРИМЕНЕНИЕ РАЗЛИЧНЫХ МАТЕМАТИЧЕСКИХ МЕТОДОВ ВЫДЕЛЕНИЯ ФОНА И ЭЛЕМЕНТОВ ТЕКСТА ИЗ ФОТОГРАФИЙ

**Брусилова Ирина Владимировна[1], Дедович Татьяна Григорьевна[2]**

*[1]Студент;*
*Государственный университет «Дубна»;*
*141980, Московская обл., г. Дубна, ул. Университетская, 19;*
*e-mail: briv.21@uni-dubna.ru.*

*[2]Старший научный сотрудник;*
*Объединенный институт ядерных исследований;*
*141980, Московская обл., г. Дубна, ул. Жолио-Кюри, д. 6;*
*Кандидат физико-математических наук, доцент;*
*Государственный университет «Дубна»;*
*141980, Московская обл., г. Дубна, ул. Университетская, 19;*
*e-mail: tdedovich@jinr.ru.*

*В работе проведено исследование применимости различных математических методов для сравнения изображений рукописных текстов. В качестве данных использовались фотографии конспектов университета «Дубна». Анализ фотографий показал, что присутствуют оригинальные фотографии, повторно прикрепленные фотографии и перефотографированные конспекты (в таком случае требуется удаление фона для дальнейшего сравнения). Для избавления от предметов фона сравнивались предложенная «многоступенчатая процедура» на основе сети U-Net и алгоритмы компьютерного зрения. Показано, что «многоступенчатая процедура» имеет преимущество. Предложены алгоритмы (быстрое сравнение изображений и сиамские нейросети) для дальнейшего анализа подготовленного изображения, на котором удален фон.*

Ключевые слова: компьютерное зрение, рукописный текст, *U-Net*.

**Для цитирования:**

Брусилова И. В., Дедович Т. Г. Применение различных математических методов выделения фона и элементов текста из фотографий // Системный анализ в науке и образовании: сетевое научное издание. 2025. №3. С. 1-16. EDN: EGCNSA. URL: https://sanse.ru/index.php/sanse/article/view/677.

# APPLICATION OF VARIOUS MATHEMATICAL METHODS OF EXTRACTING BACKGROUND AND TEXT ELEMENTS FROM IMAGES

**Brusilova Irina V.[1], Dedovich Tatyana G.[2]**

*[1]Student;*
*Dubna State University;*
*19 Universitetskaya Str., Dubna, Moscow region, 141980, Russia;*
*e-mail: briv.21@uni-dubna.ru.*

*[2]Senior Researcher;*
*Joint Institute for Nuclear Research;*
*6 Joliot-Curie Str., Dubna, Moscow region, 141980, Russia;*
*PhD in Physical and Mathematical Sciences, associate professor;*
*Dubna State University;*
*19 Universitetskaya Str., Dubna, Moscow region, 141980, Russia;*
*e-mail: tdedovich@jinr.ru*

*This paper presents the results of a research of the possible applications of different mathematical methods for the identification of duplicate lecture note images belonging to Dubna State University students. The analysis revealed that the duplicate images are exact copies or photographs of the same note taken under varying environmental conditions, necessitating background removal for subsequent comparison.For the purpose of background removal, a multi-step procedure (with usage of U-Net architecture) was compared with computer vision methods. The comparison revealed that the multi-step procedure showed better results. Furthermore, algorithms for subsequent analysis are proposed, including quick image comparison and Siamese neural network architecture.*

Keywords: computer vision, handwritten text, *U-Net*.

**For citation:**
Brusilova I. V., Dedovich T. G. Application of various mathematical methods of extracting background and text elements from images. *System analysis in science and education*, 2025;(3):1-16 (in Russ). EDN: EGCNSA. Available from: https://sanse.ru/index.php/sanse/article/view/677.

## *Введение*

В Государственном университете «Дубна» для обучения математическим дисциплинам используется разработанная в университете «Дубна» система *LmsDot* [1]. Система обеспечивает проведение семинарских занятий, проверку конспектов, контрольных и домашних работ. Однако проверка уникальности рукописных работ, поступающих в виде фотографий, осложняется большим объемом материалов и различиями в условиях съемки.

Предварительный анализ показал, что недобросовестные студенты либо прикрепляют чужие фотографии без изменений, либо перефотографируют чужой конспект, например, с другим фоном и другой освещенностью. Поэтому, для сравнения конспектов выделяются два подхода: простое сравнение фотографий (поиск повторно прикрепленных фотографий) и анализ рукописного текста с предварительным удалением лишних предметов с изображения (фона) и выравниванием освещенности. Простое сравнение фотографий не представляет сложности, а второй подход требует применения алгоритмов компьютерного зрения и использования нейросетей.

## *1. Применение различных математических методов выделения фона и элементов текста из фотографии*

Для выравнивания освещенности фотографий, удаления фона и выделения элементов текста применялись различные математические методы. Ниже представлены результаты применения алгоритмов компьютерного зрения и «многоступенчатой процедуры» на основе сети *U-Net* [2].

### 1.1. Применение алгоритмов компьютерного зрения для выделения элементов текста

Для сравнения рукописных текстов методами компьютерного зрения, позволяющих выравнивать освещенность, удалять фон и выделять элементы текста (слова, части слов, буквы) используется следующий алгоритм, на основе методов библиотеки *OpenCV* [3]:

1. Изображение считывается методом *imread* [4] в цветовом пространстве *RGB*.
2. Переводится в оттенки серого, методом *cv2.cvtColor*() [5].
3. Выделяются ребра, используя *cv2.Canny*() [6], на основе градиента яркости.

Интенсивность серого цвета определяется по формуле $0.299*R + 0.587*G + 0.114*B$, где $R$, $G$ и $B$ интенсивности соответствующих цветовых каналов.

На рис. 1 показаны интенсивности ребер *A*, *B* и *C*. Так как ребро *A* находится выше *maxVal*, то оно однозначно относится к ребру. Ребро *C* является ребром, так как находится между *minVal* и *maxVal* и

соединено с *A* (однозначно определенное ребро). Хотя ребро *B* находится выше *minVal*, но оно не соединено ни с одним однозначно определенным ребром, поэтому ребром не считается.

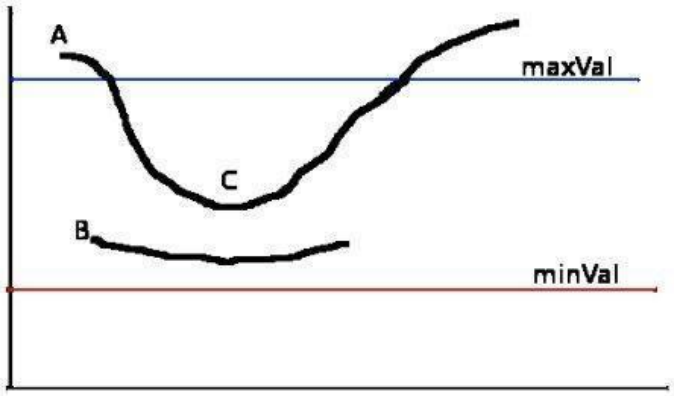
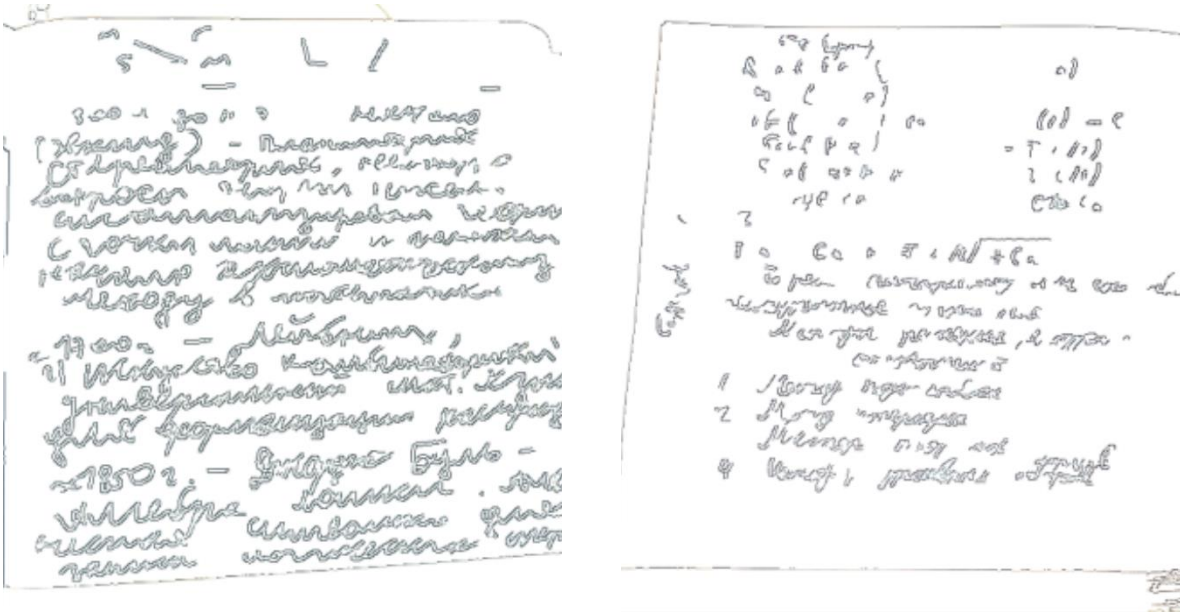



*Рис. 1. Процедура гистерезиса для определения ребер (граней)*

Применение этой процедуры показано на рис. 2.



*а)* *б)*

*Рис. 2. Пример применения процедуры выделения элементов текста с помощью компьютерного зрения*

В результате применения процедуры освещенность выровнена, буквы имеют двойной контур, на изображении присутствуют небольшое количество фоновых деталей (границы листа и блики на исходных фотографиях). На рисунке 1.1.2(а) выделенный текст хорошо читаем (с учетом особенностей почерка), а на рисунке 1.1.2(б) текст – трудночитаемый. Проблемы чтения связаны с тем, что буквы в словах выделяются не полностью (остается часть буквы). Потеря четкости текста *Loss* в данной работе определялась в процентах по формуле:

$$Loss = 100 \cdot \frac{S_{loss}}{S_{text}}, \quad (1)$$

где $S_{loss}$ – площадь текста с потерей четкости, $S_{text}$ – площадь листа с текстом. Для вычисления этих величин в графическом редакторе выделяли поле с текстом. На изображение накладывалась сетка. Это позволило рассчитать площадь листа с текстом $S_{text}$ как количество клеток сетки. Далее определялась величина $S_{loss}$ как количество клеток сетки с размытым текстом. Анализ проводился визуально.

Средняя потеря четкости текста, после применения алгоритмов компьютерного зрения, для набора из 100 фотографий составила 70%.

Условно фотографии можно разделить на два класса. Если потеря четкости *Loss* < 50%, то текст считался читаемым, в противном случае – трудночитаемым. Анализ показал, что 14% фотографий имеют читаемый текст, а 86% фотографий имеют трудночитаемый текст.

## 1.2 «Многоступенчатая процедура» на основе *U-Net* сети

Сегментация изображений представляет собой задачу, в которой требуется разделить изображение на несколько объектов или области (например, фон и лист бумаги). Для этого была разработана «многоступенчатая процедура» на основе масок, предсказанных *U-Net* сетью. Процедура состоит из следующих шагов:

1. Выделение масок (фона) на основе методов компьютерного зрения
2. Обучение нейросети на сгенерированных и доработанных вручную масках
3. Вычитание масок, предсказанных нейросетью, из исходных фотографий
4. Применение гамма-коррекции и бинаризации
5. Применение алгоритма поиска компонент связности для удаления оставшихся элементов фона и выделения элементов текста.

Ниже опишем каждый этап данной процедуры и приведем результаты анализа.

### Генерация масок на основе методов компьютерного зрения

В работе была разработана процедура выделения масок, включающих фоновые предметы:

1. Считывание изображения методом *imread* в цветовом пространстве *RGB.*
2. Сглаживание шума: (*erode*, *dilate* и *blur*).
3. *Dilate* [7] – увеличивает объект.
4. *Erode* [7] – уменьшает объект.
5. *Blur* [7] – размывает изображение с помощью усредняющего фильтра.
6. Бинаризация *RGB* изображения.
7. Перевод в оттенки серого.

*Dilate* – это операция, которая заменяет каждый пиксель изображения локальным максимумом в области под ядром. В этом случае происходит увеличение объектов на изображении.

Для размытия границ использовалась операция *erode*(), которая заменяет каждый пиксель изображения локальным минимумом в области под ядром, тем самым происходит уменьшение объектов.

Изображение размывается с использованием свертки с гауссовским ядром, что позволяет убрать лишний шум и сгладить фотографию. Гауссово ядро второго порядка (*2D*) описывается формулой:

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}, \quad (2)$$

где $G(x,y)$ – гауссово ядро с координатами $x$ и $y$, – стандартное отклонение.

Алгоритм применялся для *RGB* изображений и для изображений в оттенках серого. Наилучшие результаты были получены для изображений в формате *RGB.*

Исследовалась возможность изменения порядка увеличения объекта (*dilate*) и уменьшения объекта (*erode*). Наилучший результат был получен в описанной выше процедуре.

Для обучения будут использоваться два набора масок: *grey*, которые были получены путем перевода из *RGB* формата в оттенки серого; *black* (бинарные маски), к *grey* был применен порог бинаризации. Для масок *grey* в 10% случаев требовалась ручная доработка, а для *black* – в 100%.

## Структура *U-Net*

Для выделения фона и листа бумаги использовалась сеть *U-Net*, структура которой представлена на рис. 3.

Цвета квадратов на рисунке обозначают основные действия: желтый цвет означает свертку с матрицей размера 3 × 3 (*Conv2D* 3 × 3) и последующую функцию активации *ReLU*. Функция *ReLU* заменяет все отрицательные значения нулем, а положительные оставляет без изменений. Красный квадрат соответствует выполнению подвыборки (*MaxPool2D* 2 × 2). Операция подвыборки извлекает максимальное значение в области изображения под ядром, что помогает снизить размерность данных, сохраняя при этом важные признаки. Синий цвет обозначает транспонированную свертку на основе матрицы 2 × 2 (*TrConv2D* 2 × 2) и последующей функции активации *ReLU*. Оранжевый цвет соответствует свертке (*Conv2D* 1 × 1) и последующей сигмоидальной функции активации (*Sigmoid*). Функция *Sigmoid* выполняет нелинейное преобразование, которое входные значения (-∞; +∞) преобразует в диапазон от 0 до 1. Сверточный слой *Conv2D* 1 × 1 объединяет все карты признаков. Каждый элемент свертки определяется как взвешенная сумма этого элемента по всем каналам. Черным пунктиром обозначается копирование (*copy*). Желто-синие прямоугольники обозначают объединение скопированных карт признаков с нисходящей ветки и результатов транспонированной свертки. Над квадратиками написаны числа, обозначающие количество каналов, где под каналами подразумеваются матрицы, определяющие различные признаки. Например, матрица, у которой единицы стоят по диагонали, определяет прямую диагональную линию.

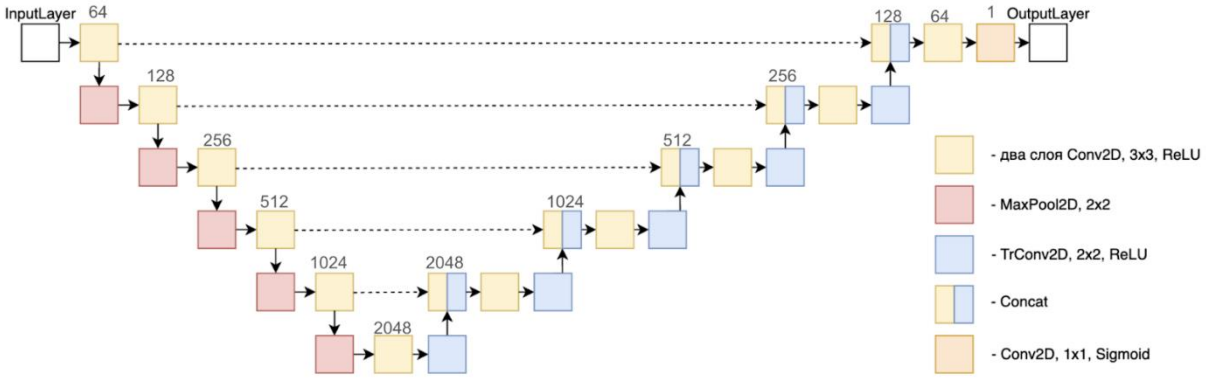



*Рис. 3. Структура U-Net сети*

## Выбор функции инициализации

Были проанализированы две функции инициализации весов (элементов матриц сверток):

1. *Glorot* [8]. Для равномерного распределения инициализация происходит в пределах [$-limit, limit$], где $limit$ вычисляется по формуле:

$$limit = \sqrt{\frac{6}{fan_{in} + fan_{out}}}, \tag{3}$$

где $fan_{\text{in}}$ – количество входных нейронов (элементы в матрицах свертки) в слое, $fan_{\text{out}}$ – количество выходных нейронов.

2. *HeNormal* [8]. Усеченное нормальное распределение с центром в нуле и со стандартным отклонением σ, которое вычисляется по формуле:

$$\sigma = \sqrt{\frac{2}{fan_{in}}}. \quad (4)$$

## Выбор функции потерь

В качестве функций потерь, которая минимизируется в процессе обучения нейросети, использовались *Binary Cross-Entropy* (*BCE*) и *Focal Cross-Entropy* (*FBCE*).

1. *BCE* [9] задается по формуле:

$$BCE = -\frac{1}{N}\sum_{i=1}^{N}\left[y_{i,true}\, ln\left(y_{i,pred}\right) + \left(1 - y_{i,true}\right) ln\left(1 - y_{i,pred}\right)\right], \quad (5)$$

где $N$ – общее количество пикселей в изображении, $y_{true}$ – истинное значение пикселя, определяющего принадлежность к классу (фон или лист бумаги), $y_{pred}$ – предсказанное значение пикселя.

2. *FBCE* [9]:

$$FBCE = -\alpha_t(1 - p_t)^{\gamma}\, ln(p_t), \quad (6)$$

где $p_t$ – вероятность принадлежности к целевому классу (к фону), $\alpha_t$ – взвешивающий коэффициент для баланса классов, $\gamma$ – модулирующий параметр, увеличивающий важность трудных примеров.

## Результаты обучения

В результате было выделено восемь вариантов гиперпараметров для обучения нейросети. В качестве масок использовались бинарные (*black*) изображения и изображения в оттенках серого (*grey*). В анализе рассматривались две функции инициализации: *HeNormal* и *Glorot*, а также две функции потерь: *Binary Cross-Entropy* (*BCE*) и *Focal Cross-Entropy* (*FBCE*).

Для оценки обучения рассматривались следующие эффективности:

1. Accuracy. Оценка доли правильных предсказаний относительно числа всех предсказаний

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \quad (7)$$

где *TP* (*True Positive*) – правильно предсказанные подходящие пиксели, *TN* (*True Negative*) – правильно предсказанные неподходящие пиксели, *FP* (*False Positive*) – неправильно предсказанные подходящие пиксели, *FN* (*False Negative*) – неправильно предсказанные неподходящие пиксели. Подходящий пиксель – это пиксель, который принадлежит фону в маске. Неподходящий пиксель – это пиксель, который не принадлежит фону.

2. *BinaryIoU*. Измеряет степень пересечения предсказанного сегмента изображения с реальным и определяется формулой:

$$BinaryIoU = \frac{TP}{TP + FP + FN}. \quad (8)$$

3. *Precision*. Показывает долю правильно предсказанных подходящих пикселей и определяется формулой:

$$Precision = \frac{TP}{TP + FP}. \quad (9)$$

4. *Recall*. Показывает долю правильно предсказанных подходящих пикселей среди всех пикселей, которые нейросеть посчитала подходящими и определяется формулой:

$$Recall = \frac{TP}{TP + FN}. \tag{10}$$

5. AUC-ROC (area under the curve ROC) это площадь под кривой ROC

Кривая *ROC* (ось *X* – ложноположительная частота *FPR*, ось *Y* – истинноположительная частота, *Recall*). Ложноположительная частота *FPR* определяется по формуле:

$$FPR\,(False\ Positive\ Rate) = \frac{FP}{FP + TN}. \tag{11}$$

Чем ближе значение *AUC* к единице, тем точнее проведена классификация. *AUC* используется для несбалансированных классов.

Метрики эффективности *Accuracy*, *Precision* и *Recall*, как правило, применяются для сбалансированных классов (приблизительно одинаковое количество пикселей для фона и для листа бумаги). Как видно из рис. 4, эффективности *Accuracy*, *Precision* и *Recall* практически не зависят от набора гиперпараметров и имеют высокое значение (≈ 0,97).

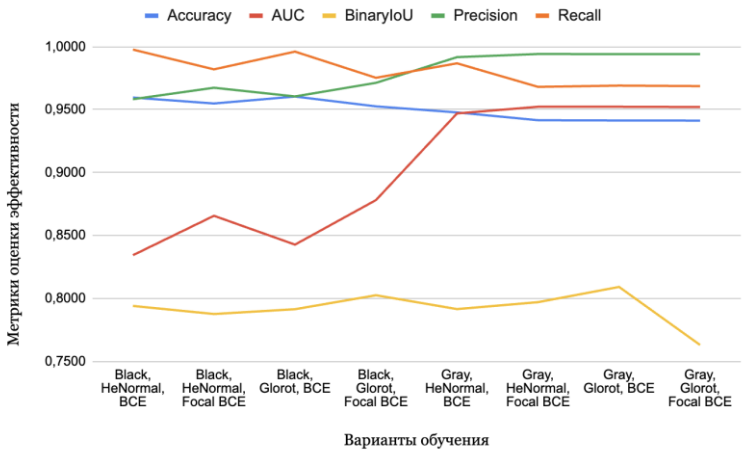



*Рис. 4. Значения оценок эффективности для различных вариантов обучения*

Метрики эффективности *AUC* и *BinaryIoU*, как правило, применяются для несбалансированных классов. Из рисунка 1.2.2 видно, что *AUC* чувствительна к типу маски (черные ≈ 0,85 и серые ≈ 0,95), а *BinaryIoU* (≈ 0,78) показывает минимальную эффективность.

Визуальный контроль показал, что для каждого набора гиперпараметров присутствуют «хорошие» (не содержат текст внутри бумаги) и «проблемные» (содержат текст и тени внутри бумаги) предсказания.

Результаты по качественному поведению ближе к критерию *AUC*, а результат для черных масок по величине близок к *BinaryIoU*.

## Вычитание предсказанных масок

Маски, предсказанные нейросетью *U-Net*, вычитались из исходных фотографий. Простое вычитание для проблемных случаев показало неудовлетворительный результат (рисунок 1.2.4(а), 1.2.4(б)), как для черных, так и для серых масок. Фон имеет неоднородную окраску сверху фотографии, и фотография очень сильно затемнена.

Поэтому была разработана процедура (см. рис. 5), которая показала лучший результат (рис. 6 (в), 6 (г)).

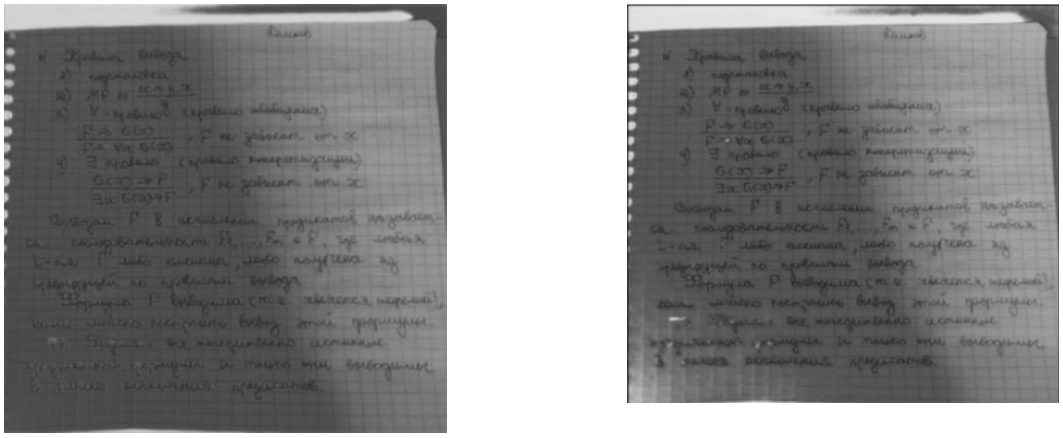
В начале определялся коэффициент $k = 2 * mean\_pred - median\_pred$, где *mean_pred* – среднее значение интенсивностей пикселей предсказанной маски, а *median_pred* – медианное значение интенсивностей пикселей предсказанной маски.

Если значения пикселя предсказанной маски меньше *k*, то соответствующий пиксель в исходной фотографии заменяется на среднее значение пикселей исходной фотографии.

```
1. k = 2 * mean_pred - median_pred
2. for i in range(512):
3.     for j in range(512):
4.         if(pred[i][j] < k):
5.             photo[i][j] = mean_photo
```

*Рис. 5. Модернизированная процедура вычитания предсказанных масок*

После применения обновленной процедуры фон более равномерный как для серых, так и для черных масок. При использовании предсказаний, полученных на серых масках, на итоговых фотографиях пропадает чуть больше текста, чем при использовании предсказаний, полученных на черных масках.



*а)* *б)*

*в)* *г)*

*Рис. 6. Пример простого (а – для черных, б – для серых масок) и модернизированного (в – для черных, г – для серых масок) вычитания*

## Использование гамма-коррекции и бинаризации

Изображения, полученные после модернизированного вычитания предсказанных масок, имеют в проблемных случаях сильно неравномерное освещение. Для устранения этого недостатка была проведена гамма-коррекция. Интенсивность каждого пикселя изменялась по формуле:

$$I_{new} = I_{tmp}^{\gamma}, \tag{12}$$

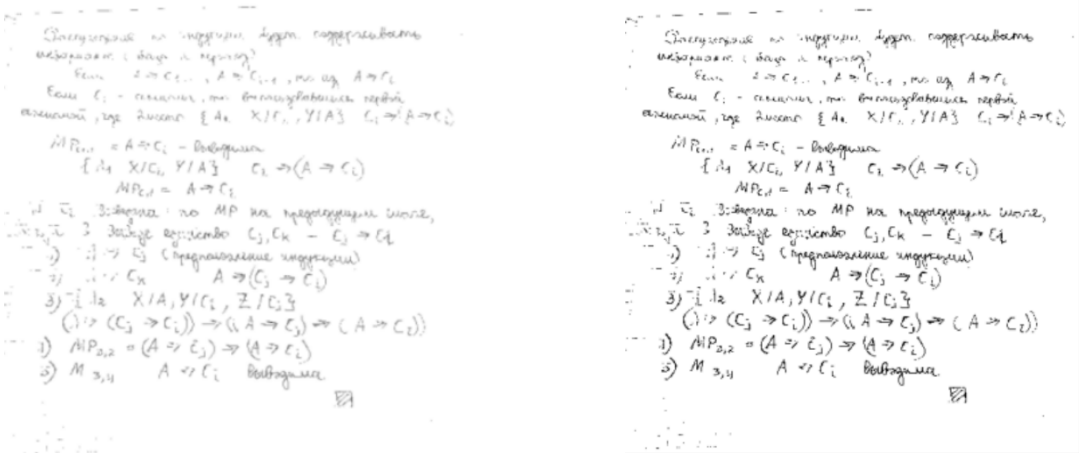
где $I_{new}$ – изображение, полученное после применения гамма-коррекции, $I_{tpm}$ – исходное изображение, – коэффициент. Наилучшие результаты получены при = 0,29.

Далее была проведена процедура бинаризации. Для каждой фотографии выполнялись следующие действия: если интенсивность пикселя больше порога, то значение интенсивности приравнивалось к единице, если меньше порога, то к нулю. Величина порога $threshold$ выбиралась из анализа ста фотографий. Рассматривались различные значения величины $threshold$. Анализ показал, что наилучшие результаты (читаемость текста и отсутствие лишних предметов) получены, если значение порога рассчитывать по следующей формуле:

$$threshold = 2 \cdot mean - median, \tag{13}$$

где $threshold$ – порог бинаризации, $mean$ – среднее значение интенсивности пикселей фотографии, $median$ – медианное значение фотографии.

В результате был получен двумерный массив, состоящий из нулей и единиц. Изображение фотографии после применения гамма-коррекции и бинаризации показано на рис. 7.



*а)* *б)*

*Рис..7. Изображения после применения гамма-коррекции с коэффициентом 0,29 (а) и последующей бинаризации (б)*

Использование гамма-коррекции и последующей бинаризации изображения позволило получить белый фон и текст. Однако, применение этой процедуры привело к частичной потере четкости текста. Потеря четкости текста *Loss* определялась в процентах по формуле (1).

Для вычисления этих величин в графическом редакторе выделяли поле с текстом. На изображение накладывалась сетка. Это позволило рассчитать площадь листа с текстом $S_{text}$ как количество клеток сетки. Далее определялась величина $S_{loss}$ как количество клеток сетки с размытым текстом. Подготовка листа с текстом для расчета этих величин показана на рис. 8. На изображение наложена

сетка с 56 клетками. Величина потери четкости текста *Loss* = 100*12/56 = 21%. Анализ проводился визуально.

*Рис. 8. Изображения с нанесенной сеткой для подсчета потери четкости текста*

Далее фотографии были разделены на два класса. Если потеря четкости Loss < 50%, то текст считался читаемым, в противном случае – трудночитаемым. Средняя потеря четкости текста, после применения «многоступенчатой процедуры», для набора из 100 фотографий составила 38%. Стоит отметить, что оригинальные фотографии в 20% случаев были плохого качества (присутствовали сильные перепады освещенности и, вследствие этого, плохо читаемые слова). Этот факт сильно снизил общую эффективность.

## Применение алгоритма по поиску компонент связности для удаления оставшихся элементов фона и выделения элементов текста

Отметим, что после всех проведенных процедур большинство элементов фона удалено, но малая часть их присутствует. Для удаления оставшихся элементов фона применен алгоритм поиска компонент связности.

Массив (изображение), состоящий из нулей и единиц, рассматривался как граф (*G*). Вершины графа (*V*) – это элементы массива (пиксели в контексте изображения), имеющие значение, равное нулю. Вершины *u* и *v* считаются смежными (соединенные ребром (*u, v*)), если они имеют общую границу (вертикальную, горизонтальную или по диагонали). У одной вершины, может быть, до восьми смежных вершин (меньше восьми для вершин на границе, у которых отсутствует часть смежных вершин). Компонента связности графа *G* – это максимальный связный подграф *G(U)*, порожденный множеством U ⊆ V(G) вершин. В этом подграфе для любой пары вершин $u, v \in G$ существует (*u, v*) – цепь, и для любой пары вершин $u \in U, w \notin U$ не существует (*u, w*) – цепи.

Поиск компонент связности выполняется алгоритмом поиска в глубину (*dfs*).

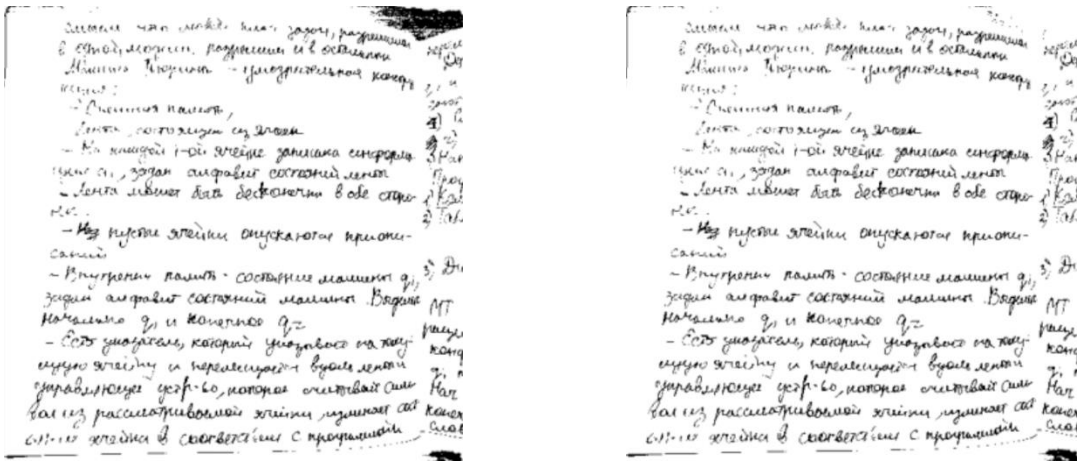
1. Начиная с первой вершины графа (первый левый верхний черный пиксель) производится поиск в глубину смежных вершин. Исходной вершине и всем найденным присваивается порядковый номер 0.
2. Пока остаются непосещенные вершины (пиксели, не вошедшие в компоненту связности) выполняется следующий шаг. Начиная с первой непосещенной вершины производится поиск в глубину смежных вершин. Исходной вершине и всем найденным присваивается порядковый номер *i*+1, где *i* – номер последней найденной компоненты связности.

Для каждой компоненты связности вычислялось количество пикселей, входящих в нее (размер компоненты). Предполагается, что элементы фона входят в компоненты связности, имеющие большое количество пикселей. Однако для различных фотографий количество компонент связности, которые соответствуют фону, отличаются. Была разработана процедура, учитывающая эти особенности.

Компоненты связности, которые нужно удалить определяются по следующему алгоритму для каждой фотографии:

1. Определяется среднее значение количества пикселей (*mean*) в компонентах связности для фотографии. Присутствие компонент связности, имеющих количество пикселей меньшее чем *mean*, приводит к малому среднему значению. Поэтому далее вычислялось среднее значение (*mean_new*) без учета таких компонент связности.
2. На основе этих двух величин вычисляется коэффициент $k = (mean + mean\_new) * 4$.
3. Компоненты связности сортируются в порядке убывания их размера и формируем список *ordered*.
4. На основе *ordered* составляется список *differences*, содержащий разности между двумя соседними элементами *ordered*.
5. В списке *ordered* сохраняются элементы, значения которых больше *k*. Такое же количество элементов сохраняется в списке *differences*.
6. В списке *differences* находится последний элемент (*last*), который удовлетворяет условию: (*differences*[*i*] > *mean_new*).
7. Компоненты связности до элемента *last* удаляются из изображения.

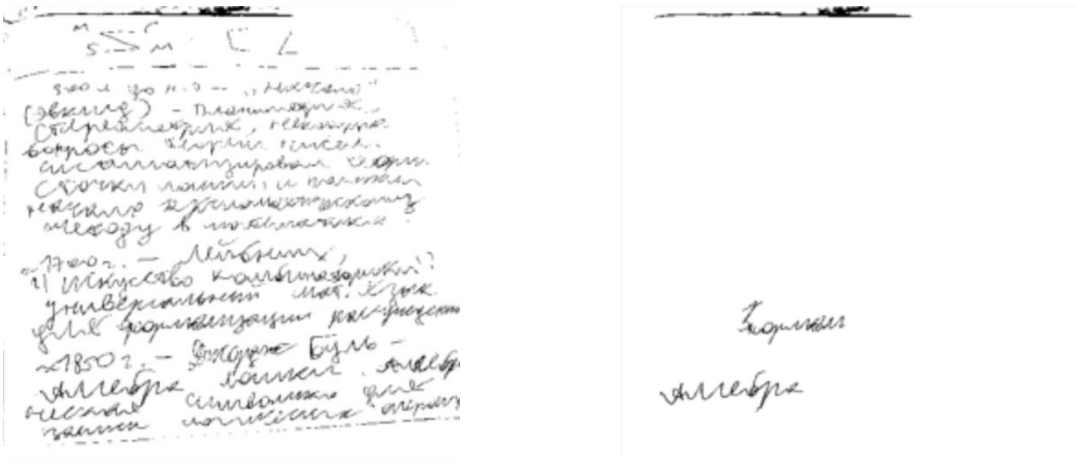
На рис. 9 показаны результаты процедуры удаления остатков фона с использованием компонент связности. На рис. 9 (а) представлено изображение после бинаризации и гамма-коррекции. Вверху и внизу справа видны остатки фона. Результат удаления лишних предметов, согласно процедуре, представлен на рис. 9 (б). Видно, что удален большой фоновый предмет в правом верхнем углу.



*а)* *б)*

*Рис. 9. Изображение после бинаризации и гамма-коррекции, а), изображение после удаления компоненты связности б)*

Удаление компонент связности на основе описанной процедуры в некоторых случаях приводило к удалению слов. Анализ показал, что в 5% случаев были удалены от 1 до 3 слов. На рисунке 10 приведен пример изображения, демонстрирующего потерю двух слов (на рисунке 10 (б) приведены удаленные компоненты связности, среди которых есть два слова).



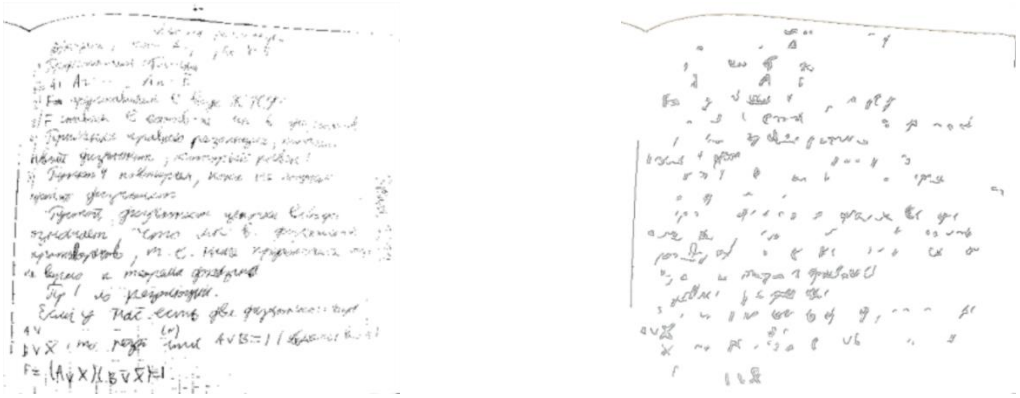
*а)* *б)*

*Рис. 10. Изображение после гамма-коррекции и бинаризации (а) и удаленные компоненты связности (б)*

### 1.3 Сравнение результатов удаления фона и выделения элементов текста методами компьютерного зрения и «многоступенчатой процедурой» на основе сети *U-Net*

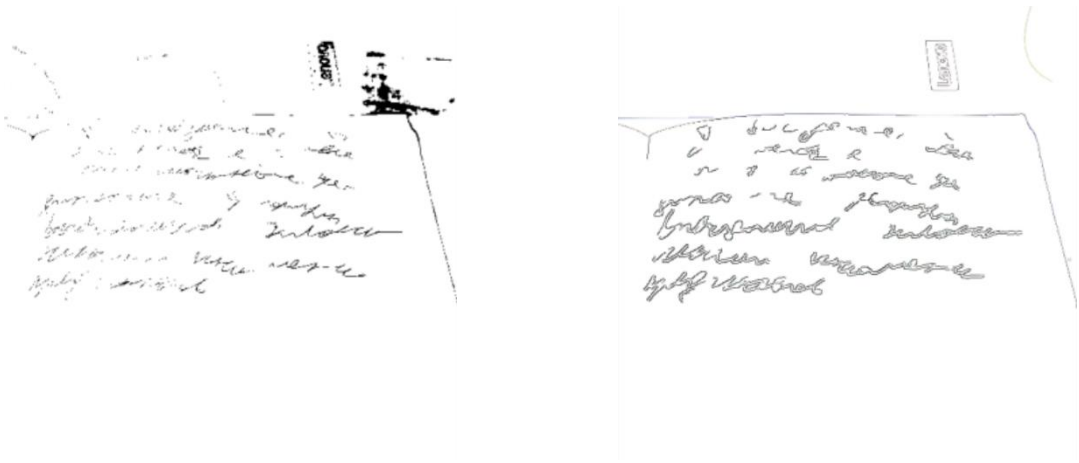
Сравнение итоговых изображений проводилось на основе анализа читаемости текста и количества оставшихся предметов фона.

Для рассматриваемых процедур потеря четкости текста определялась на основе величины *Loss*. Изображения были условно разделены (см. раздел 1.1 для алгоритмов компьютерного зрения и раздел 1.2 для «многоступенчатой процедуры») на читаемые и трудночитаемые. Для «многоступенчатой процедуры» доля читаемых фотографий составила 60%, а для алгоритмов компьютерного зрения 14%. Читаемые и трудночитаемые изображения для двух процедур не всегда совпадали. На рис. 11 показано преимущество «многоступенчатой процедуры», а на рисунке 12 показано преимущество применения алгоритмов компьютерного зрения. Рассматривался набор, состоящий из ста фотографий. В 69% случаев читаемость текста выше на изображениях, полученных после применения «многоступенчатой процедуры».



*а)* *б)*

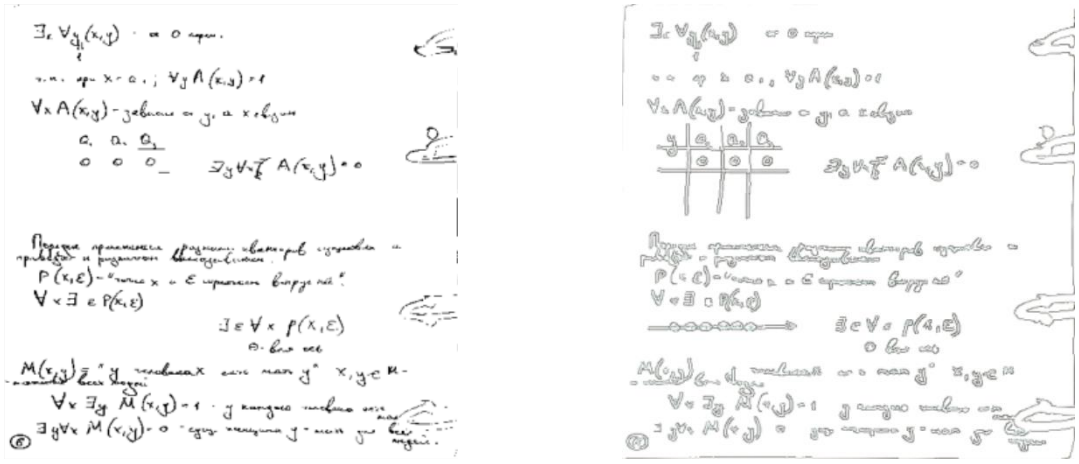
*Рис. 11. Изображение после «многоступенчатой процедуры» (а) и алгоритмов компьютерного зрения (б). Преимущество «многоступенчатой процедуры»*



*а)* *б)*

*Рис. 12. Изображение после «многоступенчатой процедуры» (а) и алгоритмов компьютерного зрения (б). Преимущество алгоритмов компьютерного зрения*

Опишем процедуру сравнения алгоритмов на основе подсчета количества оставшихся предметов фона. На рис. 14 представлены итоговые изображения после «многоступенчатой процедуры» (см. рис. 14 (а)) и после применения алгоритмов компьютерного зрения (см. рис. 14 (б)). Для каждого изображения подсчитывались лишние предметы. На рисунке 14 (а) найдено четыре лишних предмета (кольца), а на рисунке рис. 14 (б) найдено пять предметов (кольца и линия внизу). Для данной пары изображений меньше количество предметов найдено на изображении после «многоступенчатой процедуры».



*а)* *б)*

*Рис. 14. Изображение после «многоступенчатой процедуры» (а) и алгоритмов компьютерного зрения (б)*

Для каждой фотографии подсчитывалось количество оставшихся предметов фона после каждой процедуры («многоступенчатой» и алгоритмов компьютерного зрения). Рассматривался набор, состоящий из ста фотографий. Среднее количество оставшихся предметов для «многоступенчатой процедуры» составило 3,7, а для алгоритмов компьютерного зрения 4,9. Отметим, что это малые предметы, такие как края бумаги, кольца, небольшие блики. В 66% случаев меньшее количество

предметов фона осталось на изображениях, полученных после применения «многоступенчатой процедуры».

В таблице 1 приведены сравнительные характеристики методов, описанные ранее. Из таблицы видно, что «многоступенчатая процедура» на основе сети *U-Net* показывает преимущество по таким характеристикам, как доля фотографий, имеющих читаемый текст и меньшее количество предметов фона.

*Таблица 1. Сравнительные характеристики методов*

| | «Многоступенчатая процедура» на основе сети *U-Net* | Алгоритмы компьютерного зрения |
|---|---|---|
| Средняя доля потери четкости текста | 38% | 70% |
| Доля фотографий, имеющих читаемый текст | 60% | 14% |
| Среднее количество оставшихся предметов | 3,7 | 4,9 |
| Доля фотографий, имеющих меньшее количество предметов фона | 66% | 34% |

## 1.4 Обсуждение полученных результатов и предложения для дальнейшего анализа

Далее опишем предложения для улучшения результатов «многоступенчатой процедуры» на основе *U-Net* сети и предложения для поиска перефотографированных конспектов.

### 1.4.1 Предложения для улучшения результатов «многоступенчатой процедуры» на основе *U-Net* сети

Для улучшения результатов «многоступенчатой процедуры» предложены способы, которые предположительно помогут достичь лучших результатов:

1. Обучение нейросети проводить на большем количестве фотографий.
2. Подобрать функцию потерь, учитывающую особенности задачи.
3. При проведении гамма-коррекции разбить фотографии на классы, требующие различных значений коэффициентов.

Отдельно стоит отметить, что в системе *LmsDot* рекомендуется ввести фильтр на качество прикрепляемых фотографий.

### 1.4.2 Предложения для поиска перефотографированных конспектов

В результате проведения «многоступенчатой процедуры» удалена большая часть фона, и фотография имеет белый фон, черный текст. Далее определение перефотографированных конспектов можно провести при помощи сиамских сетей и алгоритма нахождения похожих изображений. Опишем кратко каждый из этих методов.

Для дальнейшего нахождения перефотографированных конспектов возможно использование нейросетевой архитектуры сиамские сети. Входными данными будут являться фотографии конспектов студентов, после проведенной «многоступенчатой процедуры». На обучение сети подаются пары перефотографированных в различных условиях конспектов и пары разных оригинальных изображений, а также целевые метки 1 в первом случае и 0 во втором случае. Предсказание нейросети – это целевые метки, которые для пары фотографий выдает степень схожести от 0 до 1 (евклидово расстояние между векторами признаков двух изображений).

Опишем алгоритм быстрого нахождения похожих изображений. Каждая фотография сначала уменьшается до размера $2 \times 2$, а затем строится четырехмерный индекс. Первый индекс $i[0]$ определяется, как средняя яркость изображения, и задается по формуле:

$$i[0] = \frac{p[0][0] + p[0][1] + p[1][0] + p[1][1]}{4}, \quad (14)$$

где $p$ – это пиксель изображения.

Второй индекс $i[1]$ отвечает за степень яркости левой части изображения по сравнению с правой и определяется по формуле:

$$i[1] = 128 + \frac{p[0][0] - p[0][1] + p[1][0] - p[1][1]}{4}. \quad (15)$$

Третий индекс $i[2]$ показывает, насколько ярче верхняя часть изображения по сравнению с нижней и задается по формуле:

$$i[2] = 128 + \frac{p[0][0] + p[0][1] - p[1][0] - p[1][1]}{4}. \quad (16)$$

Четвертый индекс $i[3]$ отвечает за различие в пикселях по диагонали и определяется по формуле:

$$i[3] = 128 + \frac{p[0][0] - p[0][1] - p[1][0] + p[1][1]}{4}. \quad (17)$$

Таким образом, каждая фотография, после данной процедуры, будет представлять точку в четырехмерном пространстве индексов $(i[0], i[1], i[2], i[3])$.

Мерой различия между изображениями, с приблизительно одинаковыми размерами, может быть евклидово расстояние. Эта мера задается значением от 0 до 1. Если мера различия имеет значение 0, то это означает, что алгоритм нашел полностью идентичные изображения. Исследование будет направлено на поиск порога, который будет находить похожие изображения с небольшими изменениями.

## *Заключение*

В статье проведено исследование различных математических методов для определения повторно прикрепленных фотографий. В качестве исходных данных использовались фотографии конспектов студентов (3000 фотографий) университета «Дубна», размещенные в системе *LmsDot*.

Анализ фотографий показал, что присутствуют оригинальные фотографии, повторно прикрепленные фотографии и перефотографированные конспекты.

Из набора фотографий случайным образом были выбраны 120 фотографий для анализа. Предварительный анализ выявил низкое качество некоторых изображений конспектов, такие как наличие посторонних предметов, засвеченные изображения, изображения с очень низким разрешением и другие сложности. Рекомендовано в системе *LmsDot* ввести дополнительное условие на прикрепление качественных фотографий, а для проверки на присутствие перефотографированных конспектов требуется вначале разделить фон и лист бумаги.

Для разделения листа бумаги и фона использовались алгоритмы компьютерного зрения. Анализ показал, что в 10% случаев выделенный фон требует значительной ручной корректировки, а средняя и малая степень корректировки требуется в 100% случаев. В работе исследовалась возможность сети *U*-

*Net* предсказывать фон на изображениях конспектов. Были рассмотрены различные наборы гиперпараметров (вариантов обучающих масок, функций инициализации параметров и функций потерь). Установлено, что наилучшие результаты в предсказании показывает функция инициализации *HeNormal* и функция потерь *BCE*.

Для подготовки данных к проверке на присутствие перефотографированных конспектов, на основе предсказанных сетью *U-Net* масок фона, была разработана «многоступенчатая процедура». Эта процедура включала вычитание предсказанных масок, гамма-коррекцию для сглаживания неравномерности освещенности листа бумаги, использование алгоритмов из теории графов для удаления оставшихся предметов фона.

Проведено сравнение результатов «многоступенчатой процедуры» и алгоритмов компьютерного зрения. Показано, что для «многоступенчатой процедуры» количество фотографий с читаемым текстом в 4,28 раза больше, а количество фотографий, имеющих меньшее количество оставшихся предметов фона больше в 1,94 раз. Средняя потеря четкости текста, после применения «многоступенчатой процедуры», для набора из 100 фотографий составила 38%, а после применения алгоритмов компьютерного зрения – 70%.

Предложены методы (алгоритм быстрого распознавания фотографий и сиамские нейросети) для дальнейшего анализа текста, подготовленного «многоступенчатой процедурой».

## *Список источников*


1. Кочешков А. Д., Смирнов Д. П., Дедович Т. Г. Разработка информационной системы контроля и оценки знаний студентов по математическим дисциплинам в университете «Дубна» // Системный анализ в науке и образовании: сетевое научное издание. 2021. № 2. С. 140–150. URL: http://sanse.ru/download/442. (дата обращения: 25.06.2025).
2. Segmentation models : [U-Net]. Pavel Iakubovskii, 2025. URL: https://smp.readthedocs.io/en/latest/models.html#unet free (access date: 25.06.2025).
3. OpenCV - Open Computer Vision Library. OpenCV team, 2025. URL: https://opencv.org/ (access date: 25.06.2025).
4. OpenCV: Image file reading and writing. URL: https://docs.opencv.org/4.x/d4/da8/group__imgcodecs.html (access date: 25.06.2025).
5. OpenCV: Color Space Conversions. URL: https://docs.opencv.org/3.4/d8/d01/group__imgproc__color__conversions.html (access date: 25.06.2025).
6. OpenCV: Miscellaneous Image Transformations. URL: https://docs.opencv.org/4.x/d7/d1b/group__imgproc__misc.html (access date: 25.06.2025).
7. OpenCV: Image filtering. URL: https://docs.opencv.org/4.x/d4/d86/group__imgproc__filter.html (access date: 25.06.2025).
8. Module: tf.keras.initializers | TensorFlow v2.16.1 // TensorFlow : [platform for machine learning and neural networks]. URL: https://www.tensorflow.org/api_docs/python/tf/keras/initializers (access date: 25.06.2025).
9. Module: tf.keras.loses | TensorFlow v2.16.1 // TensorFlow : [platform for machine learning and neural networks]. URL: https://www.tensorflow.org/api_docs/python/tf/keras/losses (access date:25.06.2025).