



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

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

Кафедра информатики и программирования

РЕФЕРАТ

Компьютерное зрение

студентов 1 курса 141 группы

направления 02.03.03 Математическое обеспечение и администрирование
информационных систем

факультета компьютерных наук и информационных технологий

Кучишкина Александра Сергеевича

Королькевича Ильи Дмитриевича

Саратов 2021

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	3
ИСТОРИЯ КОМПЬЮТЕРНОГО ЗРЕНИЯ	4
ОБЛАСТИ ПРИМЕНЕНИЯ КОМПЬЮТЕРНОГО ЗРЕНИЯ	5
ПОЧЕМУ КОМПЬЮТЕРНОЕ ЗРЕНИЕ ТАК СЛОЖНО?	6
ПРЕДСТАВЛЕНИЕ ИЗОБРАЖЕНИЯ.....	7
Окна	7
МЕТОДЫ РАБОТЫ С ИЗОБРАЖЕНИЯМИ	9
Линейные фильтры	9
Нелинейные фильтры	9
Пороговая обработка	10
Дилатация и эрозия	10
ПРИМЕР ПРИМЕНЕНИЯ ПРОСТЕЙШИХ ФУНКЦИЙ.....	13
ЗАКЛЮЧЕНИЕ	16
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ.....	17

ВВЕДЕНИЕ

С развитием IT-технологий и появлением новых возможностей всё больше людей начинают стремиться к повышению производительности путём использования информационных технологий. Места применения могут быть самыми разными: от ускорения производства на заводе с помощью автоматизацию большинства процессов до проверки качества еды с помощью нейронных сетей и компьютерного зрения.

В настоящее время появляется очень много задач, в которых необходима имитация человеческого зрения и мышления. Почему же это происходит. Все довольно просто. Автоматизация процессов зачастую требует таких возможностей, ведь это может очень сильно упростить жизнь людям, или же улучшить качество какого-либо механизма.

В наше время компьютерное зрение может решать множество задач: Классификация объектов, Идентификация объектов, Распознавание символов, Отслеживание объектов, Определение положения камеры в трехмерном пространстве, Восстановление изображений и видео. Все эти возможности используются для решения множества различных задач, от оцифровки текста до обнаружения опухолей по рентгенографии и самостоятельного выбора цели ракетами.

Так что же такое Компьютерное зрение? Компьютерное зрение это подраздел Искусственного интеллекта, который позволяет компьютерам извлекать полезную информацию из цифровых изображений, видео и других цифровых визуальных данных, а также создавать какие-то выводы на основе этой информации. Из этого мы понимаем, что Искусственный интеллект позволяет компьютерам думать, а Компьютерное зрение позволяет им видеть, отмечать детали и понимать.

Компьютерное зрение, как наука, продолжает развиваться, и делает это она семимильными шагами. Что говорит нам о востребованности и актуальности этой области программирования.

Компьютерное зрение, не смотря на то, что это относительно новая наука, очень емкая область, так что в рамках данной работы решено поставить цель – разобраться в основных аспектах науки, таких как представление изображения и основные методы обработки и работы с ними.

ИСТОРИЯ КОМПЬЮТЕРНОГО ЗРЕНИЯ

Ученые и инженеры уже 60 лет пытаются разработать методы, которые позволят компьютерам видеть и понимать визуальную информацию. Все началось в 1959 году, когда нейрофизиолог показывал коту некоторое количество изображений, чтобы понять реакцию его мозга. Он заметил, что в первую очередь он реагировал на грани и линии, это означало что обработка визуальной информации начинается с простых форм, таких как прямые грани.

Примерно в это же время была разработана первая технология сканирования изображений, которая позволяла оцифровывать изображения. Следующий этап был достигнут в 1963 году, когда смогли преобразовать двухмерное изображение в трехмерную фигуру. В 1960х годах Искусственный интеллект выделили как отдельное научное направление, что отметило начало пути до решения проблем Компьютерного зрения, за счет Искусственного интеллекта. В 1974 году была разработана технология оптического распознавания символов, которая должна была распознавать печатный текст любого шрифта. А также технология интеллектуального распознавания символов, которая была призвана распознавать рукописный текст. С тех пор эти технологии нашли свое применение в обработке документов, машинном переводе, распознавании номерных знаков и во многих других задачах.

В 1982 году нейробиолог Дэвид Марр установил, что зрение работает иерархично и ввел алгоритмы для распознавания граней, углов, кривых и простых форм. В это же время, специалист в области информатики Кунихико Фукушима разработал искусственную нейронную сеть, которая должна была распознавать образы. Эта сеть, названная Неокогнитроном, включала в себя свертываемые слои в нейронной сети.

С 2000 года, изучение было сфокусировано на распознавании объектов, и в 2001 году появилось первая программа распознавания лица в реальном времени. Также в то время стандартизировали формат визуальных наборов данных. В 2010 году стал доступен ресурс ImageNet, который предоставляет наборы визуальных данных. Благодаря таким ресурсам, стало возможным довольно точное обучение моделей, используемых для компьютерного зрения.

ОБЛАСТИ ПРИМЕНЕНИЯ КОМПЬЮТЕРНОГО ЗРЕНИЯ

Одним из наиболее важных применений является обработка изображений в медицине. Эта область характеризуется получением информации из видеоданных для постановки медицинского диагноза пациентам. В большинстве случаев, видеоданные получают с помощью микроскопии, рентгенографии, ангиографии, ультразвуковых исследований и томографии. Примером информации, которая может быть получена из таких видеоданных является обнаружение опухолей, атеросклероза или других злокачественных изменений. Также примером может быть измерение размеров органов, кровотока и т. д. Эта прикладная область также способствует медицинским исследованиям, предоставляя новую информацию, например, о строении мозга или качестве медицинского лечения.

Другой прикладной областью компьютерного зрения является промышленность. Здесь информацию получают для целей поддержки производственного процесса. Примером может служить контроль качества, когда детали или конечный продукт автоматически проверяются на наличие дефектов. Другим примером является измерение положения и ориентации деталей, поднимаемых рукой робота.

Военное применение является, пожалуй самой большой областью компьютерного зрения. Очевидными примерами являются обнаружение вражеских солдат и транспортных средств и управление ракетами. Наиболее совершенные системы управления ракетами посылают ракету в заданную область, вместо конкретной цели, а селекция целей производится, когда ракета достигает заданной области, основываясь на получаемых видеоданных. Современное военное понятие, такое как «боевая осведомленность», подразумевает, что различные датчики, включая датчики изображения, предоставляют большой набор информации о поле боя, которая может быть использована для принятия стратегических решений. В этом случае, автоматическая обработка данных используется, чтобы уменьшить сложность или увеличить надежность получаемой информации.

Одними из новых областей применения являются автономные транспортные средства, включая подводные, наземные, воздушные. Уровень автономности изменяется от полностью автономных до транспортных средств, где системы, основанные на компьютерном зрении, поддерживают водителя или пилота в различных ситуациях. Полностью автономные транспортные средства используют компьютерное зрение для навигации, то есть для получения информации о месте своего нахождения, для создания карты окружающей обстановки, для обнаружения препятствий. Они также могут быть использованы для определённых задач, например, для обнаружения лесных пожаров. Примерами таких систем могут быть система предупредительной сигнализации о препятствиях на машинах и системы автономной посадки самолетов. Некоторые производители машин демонстрировали системы автономного управления автомобилем, но эта технология все ещё не достигла того уровня, когда её можно запустить в массовое производство.

ПОЧЕМУ КОМПЬЮТЕРНОЕ ЗРЕНИЕ ТАК СЛОЖНО?

Компьютерное зрение является очень сложной областью исследований. Исследовательских проблем, которые были бы решены полностью, почти нет. Основная проблема в том, что человеческая зрительная система слишком хороша при решении множества задач. Например, человек может распознавать лица при любом освещении, ракурсе, выражении лица и т.д. Нам даже не сложно узнать человека на фото, сделанном много лет назад. А также у нас очень большой запас запоминаемых лиц, настолько большой, что кажется безграничным. И на данный момент невозможно создать автономную систему, с такими возможностями.

В итоге можно выделить две проблемы:

- Как выделить и представить огромное количество человеческих знаний в компьютере в таком виде, чтобы эту информацию было легко извлекать
- Как выполнять огромное количество вычислений, чтобы задача могла быть решена в реальном времени

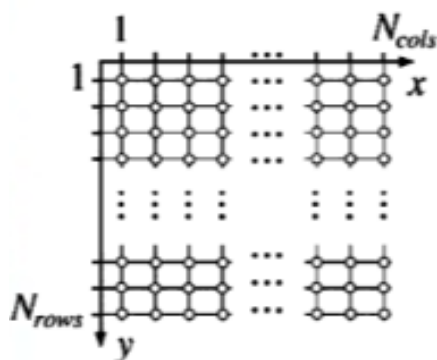
Так, можно понять, что пока почти все упирается в возможности компьютеров. Но что разработают быстрее, оптимальные методы работы или компьютеры с небывалой производительностью, неизвестно.

ПРЕДСТАВЛЕНИЕ ИЗОБРАЖЕНИЯ

Цифровое изображение определяется посредством дискретизации непрерывных аналоговых данных. Оно состоит из прямоугольного массива пикселей (x, y, u) , где (x, y) – целые числа, определяющие местоположение пикселя, и u – *отсчет* в этой точке. Точки (x, y) образуют сетку. Формально изображение I определено на прямоугольном множестве Ω , которое называется *носителем* I :

$$\Omega = \{(x, y) : 1 \leq x \leq N_{cols} \wedge 1 \leq y \leq N_{rows}\} \subset \mathbb{Z}^2$$

и содержит узлы сетки, или местоположения пикселей.



Подразумевается левосторонняя система координат, как на рисунке выше. *Строка* y содержит узлы сетки $\{(1, y), (2, y), \dots, (N_{cols}, y)\}$ для $1 \leq y \leq N_{rows}$, а *столбец* x – узлы сетки $\{(x, 1), (x, 2), \dots, (x, N_{rows})\}$ для $1 \leq x \leq N_{cols}$.

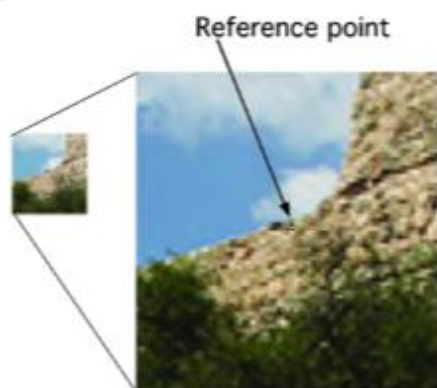
Скалярные изображения. Значениями скалярного изображения являются целые числа: $u \in \{0, 1, \dots, 2^a - 1\}$, где 0 – черный цвет, а $2^a - 1$ – белый. Остальные значения интерполируются между черным и белым. Такое изображение называют *полутонным*. Долгое время было принято использовать $a = 8$, но недавно новым стандартом стало $a = 16$.

Бинарные изображения. В бинарном изображении пиксели могут принимать два значения: 0 = белый, 1 = черный.

Векторные и RGB-изображения. В векторном изображении, в отличие от скалярного, имеется более одного *канала* или *полосы*. Значениями изображения являются векторы $(u_1, \dots, u_{N_{channels}})$ длины $N_{channels}$. Например, цветные изображения в цветовой модели RGB имеют три канала, для красной, зеленой и синей компоненты. Значения u_i в каждом канале принадлежат множеству $\{0, 1, \dots, 2^a - 1\}$.

Окна

Окном $W_p^{m,n}(I)$ называется часть изображения I , имеющая размер $m \times n$ и расположенная относительно *начальной точки* p (некоторого пикселя). Обычно подразумевается, что $m = n$ – нечетные числа, а p – центр окна. Например, на изображении ниже представлено окно $W_{(453,134)}^{77,73}$.



МЕТОДЫ РАБОТЫ С ИЗОБРАЖЕНИЯМИ

Линейные фильтры

Фильтры сглаживания делают изображение более размытым. Это может использоваться для сглаживания контуров фигур. Чаще всего, для данной задачи используют фильтр свертки. Данный фильтр работает таким образом, что значение пикселя заменяется средним значением окружающих его пикселей. При данной операции применяется ядро заданного размера и формы. Ядро – это, по сути, матрица коэффициентов на которую умножаются окружающие пиксели.

Входное изображение Матрица

12	14	41
43	84	24
2	1	43

×

0,5	0,75	0,5
0,75	1,0	0,75
0,5	0,75	0,5

=

12 * 0,5 + 14 * 0,75 + 41 * 0,5 +
43 * 0,75 + 84 * 1,0 + 24 * 0,75 +
2 * 0,5 + 1 * 0,75 + 43 * 0,5

×
1
=
div

Результат

32,41667

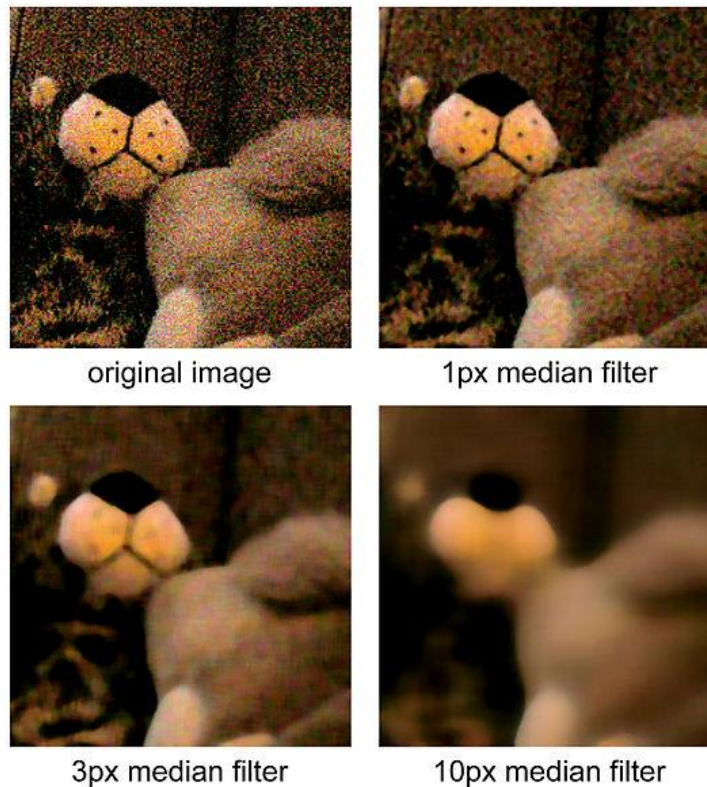
div = 6

(div – это коэффициент нормирования для сохранения средней яркости)

Простейший фильтр свертки — это фильтр усреднения. Данный фильтр использует квадратное ядро, коэффициенты в котором равны $\frac{1}{n*m}$ ($n*m$ — размерность ядра)

Нелинейные фильтры

Одним из нелинейных фильтров является медианная фильтрация. Данный метод очень эффективен при фильтрации белого шума. Одномерный медианный фильтр представляет собой скользящее окно, охватывающее нечетное количество пикселей изображения. Центральный элемент заменяется медианой всех элементов в окне. В отличие от фильтра усреднения, здесь центральный элемент заменяется одним из элементов в окне, а не вычисляется, что дает лучшее качество фильтрации.



Пример использования медианного фильтра, с разным размером окна.

Пороговая обработка

Пороговая обработка является одним из самых простых методов сегментации черно белого изображения. При пороговой обработке, каждый пиксель заменяется черным(0) пикселем если значение данного пикселя ниже заданного порога, иначе данный пиксель заменяется белым(1). Также данный метод называют бинаризацией, так как изображение приводится к двум цветам.

Дилатация и эрозия

Дилатация и эрозия – это крайне полезные методы обработки изображений для последующей работы с ними.

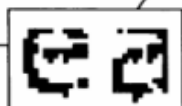
Дилатация или наращивание – это алгоритм “расширения” пикселей определенного цвета. Входными данными служат две матрицы: первая – исходное изображение, вторая – матрица, по которой будет происходить наращивание. На примере снизу видно, что алгоритм состоит в том, что в каждый удовлетворяющий условию пиксель первой матрицы подставляется центр второй матрицы.

	1	1	1	1	1	1	
			1	1	1	1	
			1	1	1	1	
		1	1	1	1	1	
			1	1	1	1	
		1	1				

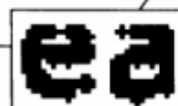
1	1	1
1	1	1
1	1	1

1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
	1	1	1	1	1	1	1
	1	1	1	1	1	1	1
	1	1	1	1	1	1	1
	1	1	1	1	1	1	1
	1	1	1	1			

Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.



Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.



Мах длина разрыва = 2 пиксела

0	1	0
1	1	1
0	1	0

Эрозия или сужение – метод, подобный дилатации, который служит для выделения объектов на изображении.

	1	1	1	1	1	1	
			1	1	1	1	
			1	1	1	1	
		1	1	1	1	1	
			1	1	1	1	
		1	1				

1	1	1
1	1	1
1	1	1

			1	1			
			1	1			
			1	1			

В результате применения эрозии все объекты, меньшие чем вторая матрица, стираются. Объекты, соединённые тонкими линиями становятся разъединёнными и размеры всех объектов уменьшаются.



$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & [1] & 1 \\ 0 & 1 & 0 \end{bmatrix}$$



$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & [1] & 1 \\ 1 & 1 & 1 \end{bmatrix}$$



$$\begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & [1] & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$

ПРИМЕР ПРИМЕНЕНИЯ ПРОСТЕЙШИХ ФУНКЦИЙ

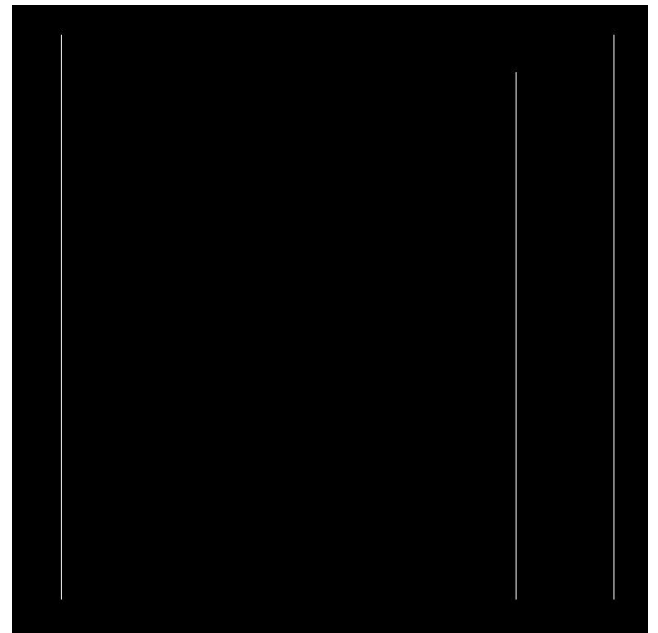
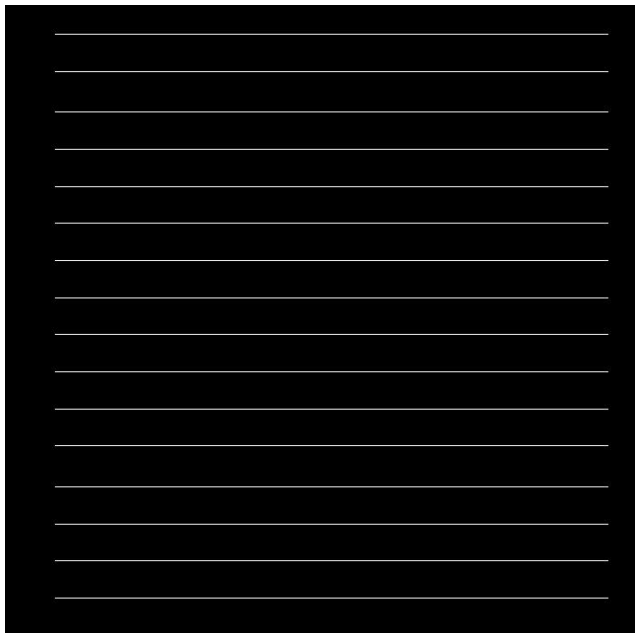
Мы решили попробовать применить простейшие функции для какой-то задачи. Остановились мы на определении ячеек таблицы. Для примера была создана анкета в виде таблицы, где в ячейках должен находиться рукописный ответ в виде цифры от 1 до 5.

Анкетирование(5 - является всегда, 4 – является часто, 3- является иногда, 2 - является редко, 1 - не является.)	
1. Всегда ли комфортно вы ощущаете себя при работе в знакомой группе	0
2. Всегда ли комфортно вы ощущаете себя при работе в незнакомой группе	1
3. Интересно ли вам работать над заданиями в группе?	2
4. Как часто прислушиваются к вашему мнению в группе?	3
5. Как часто вы прислушиваетесь к мнению других участников группы?	4
6. Можете ли вы изменить свою точку зрения при коллективном обсуждении?	5
7. Нравится ли другим ребятам работать совместно с вами в команде?	6
8. Умеете ли вы сглаживать конфликтные ситуации, возникающие при совместной работе?	7
9. Как часто вы выступаете с защитой наработок всей группы?	8
10. Всегда ли вы четко представляете конечную цель совместной работы?	9
11. Всегда ли вы четко определяете свою роль, свою задачу при продвижении к совместной цели группы?	1 0
12. Всегда ли вы добросовестно выполняете свою часть от общего задания группы?	1 1
13. Всегда ли вы стараетесь изначально самостоятельно поработать над своей частью от общего задания?	1 2
14. Всегда ли вы заинтересованы в конечном результате своей команды?	1 3

Мы пользовались языком программирования Python и библиотекой Opencv. Для начала мы приводим изображение к черно-белому, а затем используем пороговую обработку для удобства работы. В итоге мы получаем такое изображение.

Анкетирование(5 - является всегда, 4 – является часто, 3- является иногда, 2 - является редко, 1 - не является.)	
1. Всегда ли комфортно вы ощущаете себя при работе в знакомой группе	0
2. Всегда ли комфортно вы ощущаете себя при работе в незнакомой группе	1
3. Интересно ли вам работать над заданиями в группе?	2
4. Как часто прислушиваются к вашему мнению в группе?	3
5. Как часто вы прислушиваетесь к мнению других участников группы?	4
6. Можете ли вы изменить свою точку зрения при коллективном обсуждении?	5
7. Нравится ли другим ребятам работать совместно с вами в команде?	6
8. Умеете ли вы сглаживать конфликтные ситуации, возникающие при совместной работе?	7
9. Как часто вы выступаете с защитой наработок всей группы?	8
10. Всегда ли вы четко представляете конечную цель совместной работы?	9
11. Всегда ли вы четко определяете свою роль, свою задачу при продвижении к совместной цели группы?	1 0
12. Всегда ли вы добросовестно выполняете свою часть от общего задания группы?	1 1
13. Всегда ли вы стараетесь изначально самостоятельно поработать над своей частью от общего задания?	1 2
14. Всегда ли вы заинтересованы в конечном результате своей команды?	1 3

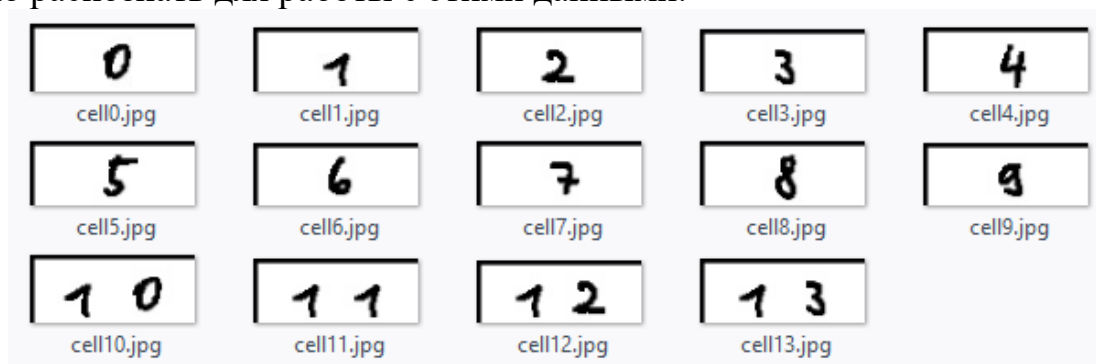
Далее применяется метод эрозии для выделения горизонтальных и вертикальных линий, а затем применяется дилатация для этих изображений. (Используется ядро (50x1) для горизонтальных линий и (1x50) для вертикальных).



После мы применяем битовое умножение для нахождения пересечений линий



Остается только найти координаты этих точек и получить изображения ячеек. В итоге мы получаем содержимое каждой ячейки, которое в дальнейшем можно распознать для работы с этими данными.



Таким образом мы смогли применить простейшие функции. В дальнейшем эту идею можно развивать и применять, например для автоматического анализа каких-либо анкет.

ЗАКЛЮЧЕНИЕ

Итак, в заключении можно отметить, что сейчас появляется все больше и больше задач для компьютерного зрения, оно становится все более обширным и захватывает все больше областей. Но на данный момент не все задачи могут быть решены. В ходе проделанной работы поставленные цели, такие как основы представления изображения, а также базовые методы обработки, были выполнены, а также был продемонстрирован пример применения компьютерного зрения.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ

1. Computer Vision: Evolution and Promise [T. S. Huang, University of Illinois at Urbana-Champaign]
2. Р. Клетте - Компьютерное зрение. Теория и алгоритмы – 2019
3. http://edu.mmcs.sfedu.ru/pluginfile.php/15283/mod_resource/content/10/Binary.pdf
4. http://neerc.ifmo.ru/wiki/index.php?title=Компьютерное_зрение
5. <https://shalaginov.com/2020/05/16/computer-vision-history/>
6. <https://www.ibm.com/topics/computer-vision>
7. <https://habr.com/ru/post/519454/>
8. <https://russianblogs.com/article/2777646555/>
9. <https://habr.com/ru/post/274725/>
10. <https://iopscience.iop.org/article/10.1088/1361-651X/abfd1a>
11. https://github.com/Liquicious/Computer-Vision-project/tree/side_branch1