

## 1. МЕТОДЫ РАСПОЗНАВАНИЯ

Мобильные роботы – это автоматические устройства, которые работают как машина, управляемая программным обеспечением. Мобильные роботы становятся все более популярными в различных сферах промышленности. Для перемещения робота, существуют такие физические элементы, как: колеса, гусеницы, ноги. Роботы, для выполнения своих задач, используют различные датчики и технологии, одним из которых является технология технического зрения, для определения окружения вокруг него, распознавания различных препятствий и перемещения в пространстве.

### 1.1 Что такое компьютерное зрение и какие задачи оно решает

**Что же такое компьютерное зрение?** Для дальнейшего анализа, нужно посмотреть на этот вопрос в качестве стандартного человеческого зрения и ответить на вопрос: Какую информацию получает человек, когда видит какое то изображение?

В качестве примера, посмотрим на рис. 1.1:



Рисунок 1.1 – Вулкан Фудзияма, Япония

Какую информацию может извлечь из этого изображения человек? Обычному человеку понятно, что это фотография природы, где виден вдали вулкан. Также очевидно, что нижняя часть изображения – это какой-то водоем. Так же, человек понимаем, что катер плывет. Как же мы это понимаем?

Он оставляет за собой характерный след. Смотря на данное изображение, человек может извлечь и обработать много информации, полагаясь на: жизненный опыт, ассоциативные связи, логическое мышление и так далее. Человек получает большой объем информации с помощью зрения. Для человека возможность видеть и распознавать объекты является естественной и привычной возможностью, что в свою очередь, является достаточно сложной задачей для компьютера.

Однако с точки зрения компьютера любое изображение – это только набор точек определенного цвета. Изображение в памяти будет представлено матрицей, где каждый элемент будет описывать цвет точки, а координаты элемента будет описывать ее местоположение. Максимум информации, которую можно извлечь без дополнительной программной обработки – размер изображения и число каналов, которыми описывается цвет.

Очевидно, что разница в восприятии изображения человеком и компьютером огромна. Техническое зрение же занимается сведением этой разницы к минимуму.

**Техническое зрение** – это технология создания таких автоматизированных систем, которые способны выполнять такие задачи, как:

- наблюдение за производственной линией;
- отслеживание ошибок и неисправностей;
- сортировка продукции и многое другое

Данная область объединяет такие сферы, как: робототехника, компьютерные сети, цифровые технологии. Так же, техническое зрение является частью искусственного интеллекта.

## **Какие задачи решает техническое зрение ?**

Основная цель технического зрения – это определение, исследование и обнаружение различных объектов. Данная технология получает необходимую информацию из изображения или серии изображений(видеопотока). Задачи могут быть самыми разными, такими как:

- Распознавание лиц
- Детектирование границ, разметок
- Определение движения объектов (Motion Tracking)
- Задачи распознавания объектов
- Распознавание дорожных знаков
- Задачи реконструкции сцены
- Восстановление изображений

Техническое зрение все больше развивается, и находит свое применение в самых разных сферах, касающихся обработки изображений и распознавания образов в бытовых, промышленных и военных сферах.

**Вывод:** В заключении главы можно сделать вывод о том, что техническое зрение является гибкой и универсальной технологией для выполнения различных решений задач как и в робототехнике, так и в повседневной жизни в целом. Были рассмотрены примеры восприятия мира как человеческим так и техническим зрением.

## 1.2 Существующие методы распознавания

Выделим основные методы распознавания:

- *Пороговое преобразование*

Одним из самых простых и начальных методов распознавания служит преобразование по бинаризации изображения по порогу. Как следует из названия, пороговое значение изображения позволяет нам применять определенный “порог”, чтобы определить, представляет ли каждый пиксель для нас интерес или нет. Следовательно, существует множество практических и полезных применений пороговой обработки изображения. Например, мы можем использовать это для создания маски, чтобы изолировать определенные части изображения. Так же, это может быть хорошим средством для уменьшения шума изображения.

Возьмем задачу, где нужно разделить фон и текст/линии так, чтобы они попали в разные группы (по признаку яркости). Пример работы автоматической бинаризации по порогу представлен на рис.1.2:



Рисунок 1.2 – Результат автоматической бинаризации по порогу

Выбор порога, по которому происходит бинаризация, во многом зависит от типа самого изображения. В частности, значение порога определяется визуально, или выбирается метод автоматического подбора порогового значения. В нашем примере, изображение было преобразовано посредством автоматического подбора порогового значения.

Обычно, бинаризация осуществляется с помощью алгоритма, который адаптивно выбирает порог(Чем мы и воспользовались в данном примере)

- *Обнаружение границ*

В данном методе используется фильтрация границ и контуров.

Контуры могут быть максимально полезны, когда мы хотим перейти от работы с изображением к работе с объектами на этом изображении. Для обнаружения границ на изображении, используется ядро и фильтр. Ядро содержит вещественные числа, которые помогают в процессе обнаружение границ.

Фильтр в свою очередь нужен, для сглаживания границ, чтобы она была близка к идеальной. На рис.1.3 представлен пример идеальной и реальной границы:



Рисунок 1.3 – Пример идеальной и реальной границы

Существует целый ряд алгоритмов(функций), решающих задачу фильтрации контуров. К основным относятся:

- Оператор Собеля (Sobel)
- Оператор Лапласа (Laplacian)
- Детектор границ Кэнни (Canny)

При использовании данных алгоритмов, лучшим считается – Детектор границ Кэнни. В детектор передается сглаженное изображение.

Основные этапы детектора границ Кэнни, которые делают его одним из лучших при распознавании объектов:

- Сглаживание
- Поиск градиентов
- Подавление не-максимумов
- Двойная пороговая фильтрация
- Трассировка областей неоднозначности

Пример работы детектора границ Кэнни приведен на рис.1.4:

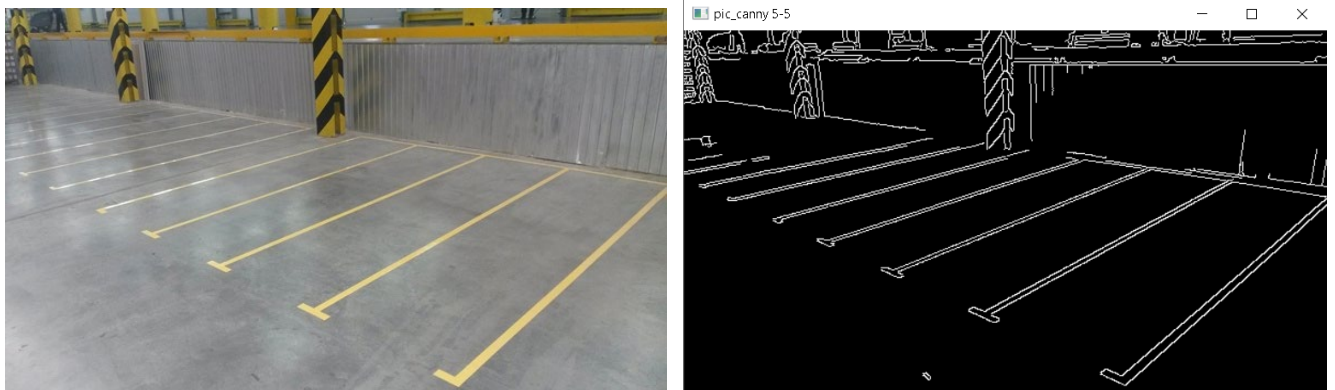


Рисунок 1.4 – Пример детектирования границ Кэнни

- *Примитивы Хаара*

Примитивы Хаара - это функции, которые представляют собой метод машинного обучения, используемый для обнаружения объектов на изображениях. Примитивы Хаара работают, обучая классификатор обнаруживать особенности в изображениях, характерные для исследуемого объекта. Примитивы Хаара действуют на изображение как свертка, пробегая по нему. На каждом прямоугольнике Хаара вычисляется параметр, равный разности суммарных интенсивностей областей рисунка и делит их на две части – белые и черные. Разновидности примитивов Хаара представлены на рис.1.5:

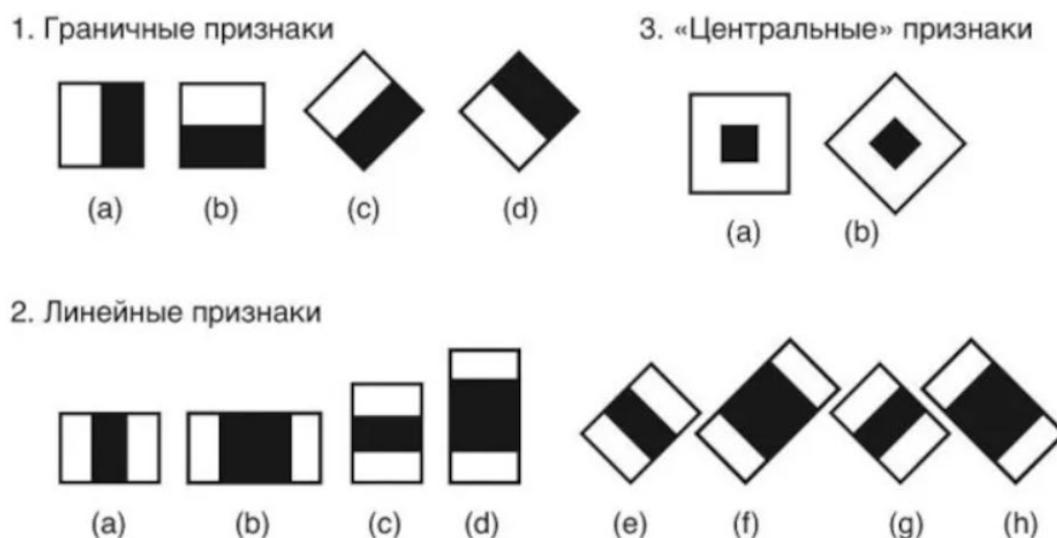


Рисунок 1.5 – Примитивы Хаара

Плюс данного метода состоит в том, что примитивы Хаара достаточно быстро вычисляются через интегральное представление изображений [1]:

$$I(x, y) = \sum_{i=0, j=0}^{i \leq x, j \leq y} I(i, j) [1]$$

Где  $(i, j)$  - яркость пикселя исходного изображения.

Примитивы Хаара использует набор простых функций, которые представляют собой повторяющиеся шаблоны черных и белых пикселей. Эти функции вычисляются в разных масштабах и местах на изображении, и примитив использует их, чтобы определить, присутствует ли объект. Есть несколько плюсов и минусов использования примитивов Хаара для обнаружения объектов:

#### Плюсы:

- Примитивы Хаара способны обрабатывать большие вариации внешнего вида объектов, такие как изменения масштаба, ориентации и освещения;
- Примитивы Хаара просты и легко вычислимы, что делает их быстрыми и эффективными;
- Примитивы Хаара способны обрабатывать изображения в режиме реального времени, что делает его пригодным для использования в приложениях, требующих быстрого отклика;

## Минусы:

- Прimitives Хаара не так точны, как некоторые другие методы обнаружения объектов, такие как методы глубокого обучения, основанные на сверточных нейронных сетях (CNN);
- Прimitives Хаара чувствительны к качеству обучающего набора данных, поэтому, если набор данных не является репрезентативным для обнаруживаемых объектов, primitives могут работать плохо;
- Прimitives Хаара способны обрабатывать изображения в режиме реального времени, что делает его пригодным для использования в приложениях, требующих быстрого отклика;
- *Поиск по шаблону*

Данный метод заключается в том, что для поиска образов используется вычисление моментов контура. Моменты контуров – это высокоуровневые характеристики, которые могут быть применены и к контуру, и к изображению, и ко множеству точек. Пусть  $I$  – это некоторое изображение. Тогда момент можно вычислить по формуле [2]:

$$m_{p,q} = \sum_{i=1}^N I(x_i, y_i) x^p y^q [2], \text{ где:}$$

$p$  и  $q$  – это некоторые коэффициенты, которые определяют, какой момент вычисляется;

$m_{p,q}$  – непосредственно значение момента;

$I(x_i, y_i)$  – один из пикселей изображения;

$N$  – размер стороны изображения;

$x^p y^q$  – некоторый коэффициент, который соответствует типу момента;

Существуют разные типы моментов. Например, для изображения можно вычислить по формуле 4.4 обычные моменты, центральные моменты и нормированные центральные моменты. Среди обычных моментов встречаются нулевые, первые, вторые и третьи:



- $m_{0,0}$  – нулевой момент – единственный;
- $m_{1,0}, m_{0,1}$  – первые моменты – два варианта;
- $m_{2,0}, m_{1,1}, m_{0,2}$  – вторые моменты – три варианта;
- $m_{3,0}, m_{2,1}, m_{1,2}, m_{0,3}$  – третьи моменты – четыре варианта.

В зависимости от того, какой момент вычисляется, значение коэффициента  $x^p y^q$  меняется. Одним из главных преимуществ использования моментов как характеристики контуров при сравнении является возможность вычислить моменты таким образом, чтобы они были инвариантны относительно поворота, переноса и масштабирования.

Чтобы применять моменты для сравнения контуров так, чтобы значения были инвариантны относительно параллельного переноса, нужно считать центральные моменты. Это можно сделать по формуле [3]:

$$\mu_{p,q} = \sum_{i=1}^N I(x_i, y_i) (x - \bar{x})^p (y - \bar{y})^q \quad [3],$$

$$\text{где } \bar{x} = m_{1,0}/m_{0,0} \text{ и } \bar{y} = m_{0,1}/m_{0,0}$$

Хоть центральные моменты и позволяют решить проблему расположения контуров в разных частях изображения, они не позволяют считать одинаковыми контуры, чей масштаб отличается. Эту проблему позволяет решить расчет нормированных центральных моментов, которые будут инвариантны относительно масштабирования. Для расчета этих моментов можно воспользоваться формулой [4]:

$$\nu_{p,q} = \frac{\mu_{p,q}}{m_{00}^{\left(\frac{p+q}{2}+1\right)}} \quad [4]$$

На рис.1.6 приведен пример нахождения звезд методом поиска по шаблону:

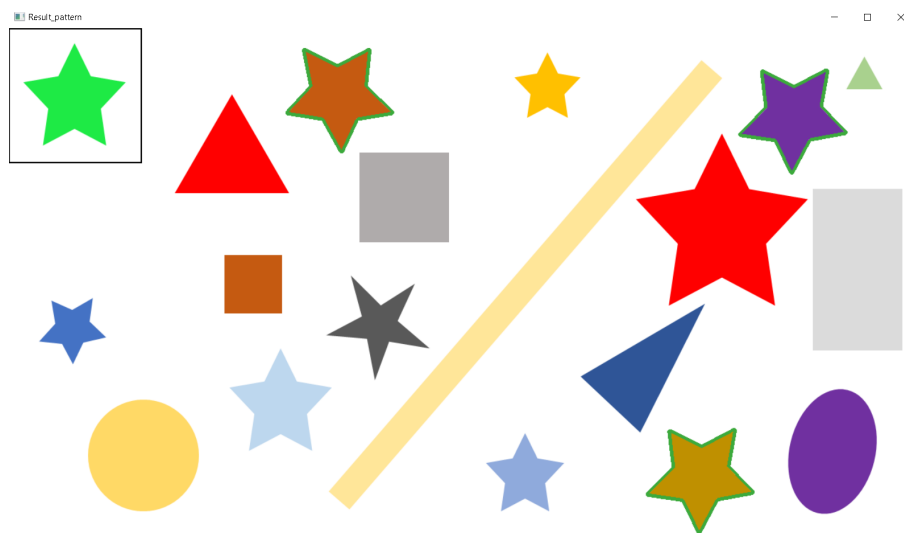


Рисунок 1.6 – метод поиска по шаблону

- *Цветовое пространство*

**Цветовое пространство** - модель для представления цвета в терминах значений интенсивности. Представляет собой специфическую организацию цветов;

Цветовое пространство – это определенная организация цветов, которая позволяет нам последовательно представлять и производить цвета. Оно определяет одно, двух, трёх, или четырехмерное пространство, чьи измерения, или компоненты, представляют собой значения интенсивности цвета. Цветной компонент также называют цветным каналом.

К примеру, **RGB-пространство** - это трехмерное цветовое пространство, в которой красный, зеленый и синий цвета складываются вместе различными способами для воспроизведения широкого спектра цветов. Визуально эти пространства часто представляют различными объемными фигурами - кубами, конусами или многогранниками. На рис.1.7 представлено цветовое пространство RGB:

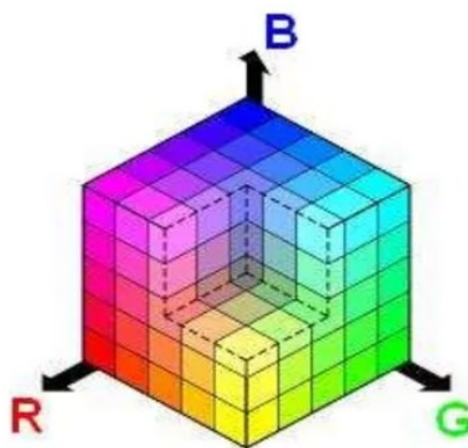


Рисунок 1.7 – цветовое пространство RGB

Цветовое пространство **HSV** разлагает яркость цвета и широко используется в алгоритмах улучшения изображения. Эта цветовая схема определяется тремя компонентами:

- **Hue** - цветовой тон;
- **Saturation** - насыщенность;
- **Value** - яркость;

**Hue** - цветовой тон. Оттенок представляет цвет. Определяется, какой «чистый» цвет исследуется. К примеру, все тени и тона цвета «красный» будут иметь одинаковый оттенок. Здесь оттенок варьируется в пределах  $0\text{—}360^\circ$ , однако иногда приводится диапазон 0-100 или 0-1.

**Saturation** - насыщенность. Насыщенность указывает диапазон серого в цветовом пространстве. Варьируется в диапазоне 0 – 100 %. Иногда значение вычисляется от 0 до 1. Когда значение равно 0, цвет становится серым, когда равно 1 – цвет является основным цветом.

**Value** – яркость цвета, которая зависит от насыщенности цвета. Варьируется в диапазоне 0 – 100 % или 0 – 1. Когда значение равно 0 – цветовое пространство будет полностью черным. С увеличением значения, цветовое пространство становится ярче и отображает различные цвета. На рис.1.8 представлено цветовое пространство HSV:

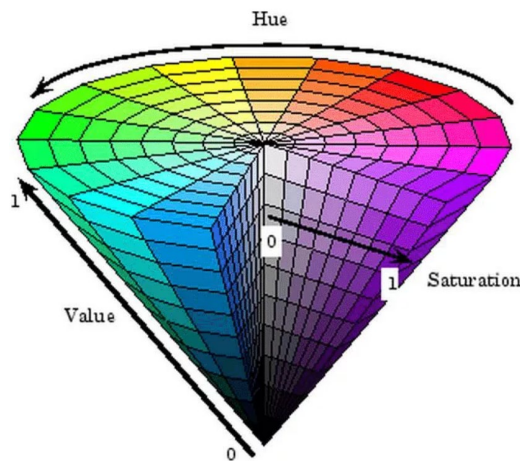


Рисунок 1.8 – цветовое пространство HSV

**Вывод:** Были рассмотрены различные существующие решения к переходу стандартного изображения к методам распознавания объектов (Пороговое преобразование, детектирование границ, поиск по шаблону и т.п). Так же, были рассмотрены влияние методов распознавания по цветовому пространству (стандартное RGB и HSV пространства).

### 1.3 OpenCV

OpenCV (Open Source Computer Vision) – это библиотека программных функций, с открытым исходным кодом, которые используются для компьютерного зрения. Данная библиотека написана на языке C++, но интерпретирована во многих других популярных языках. Она предоставляет базовые средства для решения задач технического зрения. Это помогает в решении сложных частей задачи, и одновременно дает простор для дополнения получившихся решений разработчиком. Главное достоинство OpenCV в том, что она является кроссплатформенным и бесплатным для использования по лицензии в различных целях (Учебных и коммерческих).

В нашей работе, мы будем рассматривать использование OpenCV при программировании на языке Python в среде разработки PyCharm.

Данная библиотека содержит алгоритмы для обработки изображений, распознавания образов/объектов, работы с видеоданными и др.

Библиотека OpenCV применяется:

- **В робототехнике** — для ориентирования робота в пространстве, распознавания объектов и взаимодействия с ними;
- **Медицинских технологиях** — для создания точных методов диагностики, например 3D-визуализации органа при МРТ;
- **Промышленных технологиях** — для автоматизированного контроля качества, считывания этикеток, сортировки продуктов и пр.;
- **Безопасности** — для создания «умных» камер видеонаблюдения, которые реагируют на подозрительные действия, для считывания и распознавания биометрии;
- **На транспорте** — для разработки автопилотов.

**Вывод:** В данной главе, была описана библиотека OpenCV которая является фундаментальной для решения задач технического зрения. Так же, был рассмотрен язык Python в среде разработки PyCharm, который будет использоваться в дальнейшем в нашей работе.

## 1.4 Функции OpenCV

Чтобы начать работу с библиотекой OpenCV, ее нужно импортировать командой `import cv2`. Так же, нам понадобится библиотека NumPy для работы с многомерными массивами. В чем заключаются основные функции для OpenCV:

1. **Получение изображения:** На вход подается некоторое изображение с различных камер, датчиков расстояния, ультразвуковых камер и т.д. Интенсивности света обычно соответствуют значения пикселей в одной или нескольких спектральных полосах (цветные или изображения в оттенках серого), которые также зависят от различных физических измерений, как глубина, поглощение или отражение звуковых или электромагнитных волн, ядерный магнитный резонанс. **Пример функции:** `cv2.imread()`, `cv2.VideoCapture()`
2. **Обработка изображения:** В зависимости от того, какие изображения мы получаем, делаем те или иные операции обработки: Удаления шума, сглаживание, улучшение контрастности и т.д. **Пример функции:** `cv2.blur()`, `cv2.morphologyEx()`
3. **Выделение деталей:** Проведя анализ изображения, можно выделить линии, границы углы, которые относятся к структуре, форме или движению. **Пример функции:** `cv2.findContours()`
4. **Сегментация цветов:** Сделать изображение бинаризованным(яркости 0 и 255). Так же, это позволить уменьшить размер изображения, что приведет к более быстрой дальнейшей обработки изображений. Благодаря сегментации цветов, мы обнаруживаем, какие точки и участки изображений являются важными для обработки. **Пример функции:** `cv2.threshold()`
5. **Детектирование границ:** С помощью оператора Собеля, Лапласа или детектирования границ Кэнни максимально четко выделить границы, с которыми мы будем в дальнейшем работать (Пример: Воспользоваться преобразованием Хафа для выделения линий или окружностей). **Пример функции:** `cv2.Canny()`, `cv2.Sobel`, `cv2.Laplacian` (`cv2.HoughLinesP()`, `cv2.HoughCircles()`)

**Вывод:** В данной главе, были рассмотрены основные функции OpenCV, которые служат фундаментом для методов распознавания объектов в нашей работе. Данные функции были поэтапно расписаны, как и в каком порядке они должны работать.

### 1.5 Обзор готовых решений

Одним из примеров готовых решений можно рассмотреть внедрение биометрии для клиентов банка «Сбербанк»

Биометрия - система распознавания и идентификации человека по уникальным физическим признакам организма, таким как отпечатки пальцев, радужная оболочка глаз, различных уникальностей на лице(родинки, веснушки) и т.п. В области информационных технологий биометрические данные используются в качестве формы управления идентификаторами доступа и контроля доступа. С помощью биометрии, клиенты банка «Сбербанк» могут максимально просто пользоваться услугами банка, не имея с собой физической карты. Пример технологии приведен на рис.1.9:

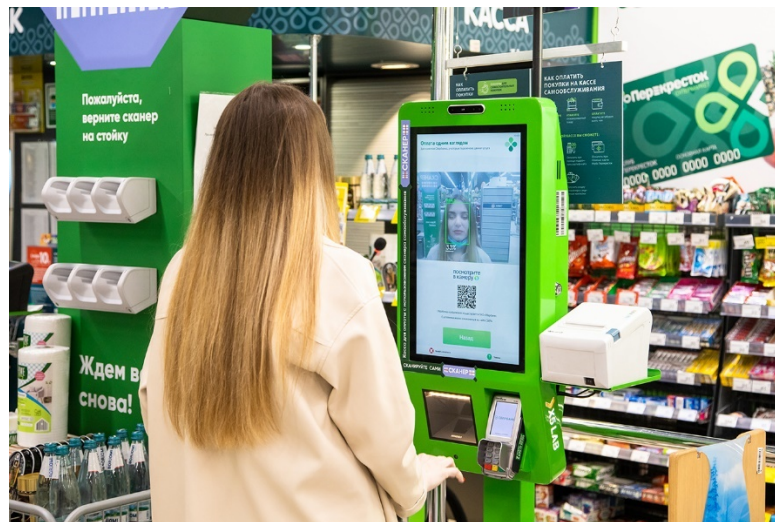


Рисунок 1.9 – Пример технологии биометрии

В плане промышленной области, компания «OMRON» создала микропроцессорный датчик технического зрения. Это компактная система обработки изображения. Датчик состоит из двух отдельных модулей - камеры со встроенным источником света и модуля обработки данных. Его можно расширить

путем подключения дополнительных контроллеров, соединяя их друг с другом в один ряд. Пример технологии приведен на рис.1.10:



Рисунок 1.10 – Пример технологии технического зрения «Omron»

Система технического зрения компании «OMRON» значительно помогает производителям увеличить мощности и сократить расходы, связанные с появлением дефектов на конечном товаре. Камеры с высокой чувствительностью и быстрой обработкой позволяют производителям наблюдать за надежными и детализированными результатами тщательных проверок на постоянной основе.

В плане дорожного движения, используется система распознавания автомобильных номеров. Вот уже несколько лет, данная система упрощает многие задачи, как и в плане нарушителей, так и в плане комфорта человека. Система распознает номера автомобилей за счет анализа видеоданных, поступающих с камер, и сохраняет информацию обо всех проехавших автомобилях в базу данных: дата/время проезда, направление проезда, изображения автомобиля, изображения номера автомобиля и т.д. Сопоставляя распознанные номера со списками и параметрами доступа, система дает команды внешним устройствам - шлагбаумам, воротам, светофорам(в плане нарушителей: отправляет данные нарушителя в базу данных ГИБДД, откуда формируется отчет



со штрафом и отправляется владельцу). Пример технологии системы распознавания автомобильных номеров представлен на рис.1.11:

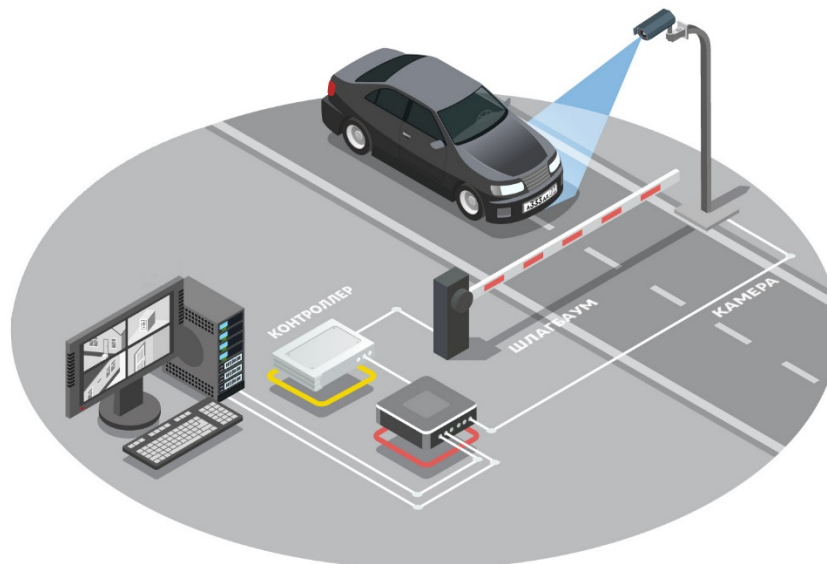


Рисунок 1.11 – Пример технологии распознавания системы распознавания автомобильных номеров

**Вывод:** В данной главе, был сделан обзор на уже существующие готовые решения, написанные с помощью библиотеки OpenCV, которые используются в повседневной жизни (Как пример: Биометрия в банкоматах и система распознавания автомобильных номеров в паркингах и автомобильных дорогах)