

Geekbrains

**Исследование систем машинного зрения для
использования в проекте «Отслеживание пешеходного
трафика»**

IT-специалист:

Суриков М.А.

Нижний Новгород

2024

1. Введение
2. Нейронные сети и машинное зрение
 - 2.1. Нейронные сети – общая информация
 - 2.2. Машинное зрение – общая информация
3. Проект – общая информация
 - 3.1. Описание проблемы
 - 3.2. Возможные решения, плюсы-минусы
 - 3.3. Выбор модели
 - 3.4. Определение конечного результата, определения контрольных параметров успешной реализации
 - 3.5. Выбор оптимального оборудования
4. Создание нейросети
 - 4.1. Первичный код
 - 4.2. Тестирование системы, результаты
5. Оценка эффективности проекта
6. Заключение
7. Список литературы

Введение.

Компьютерное зрение — это область искусственного интеллекта, которая обучает компьютеры интерпретировать и понимать визуальный мир. Используя цифровые изображения с камер и видео, и модели глубокого обучения, машины могут точно идентифицировать и классифицировать объекты - а затем реагировать на то, что они "видят".

Компьютерное зрение распространилось в самых различных сферах: считывание штрихкодов, системы видеонаблюдения, системы управления автомобилями, медицинские системы анализа изображений, технологии дополненной и виртуальной реальности и во многих других областях. Машинное зрение — это попытка смоделировать механизм получения и обработки визуальной информации в человеческом мозге.

На сегодняшний день бизнес по аренде недвижимости для коммерческих нужд использует различные системы для получения статистики, необходимой для более гибкого выбора клиентами таких решений. Однако получения информации о пешеходном трафике до сих пор производится устаревшими способами, опросами вручную, по обрывочной статистике или даже по ощущениям. Такой мощный инструмент как нейронные сети и машинное зрение мало используется, или не используется вообще. Данную ситуацию возможно решить внедрением новых технологий.

После анализа информации в интернете и личного опроса участников бизнес-процесса по аренде/покупке коммерческих помещений, что данная тема крайне актуальна на текущий момент.

Цель работы: повышение эффективности выбора коммерческих помещений всеми сторонами бизнес-процесса до 90% путем получения качественной статистики в полуавтоматическом режиме или полностью автоматическом режиме.

Для выполнения задачи необходимо пройти следующие этапы:

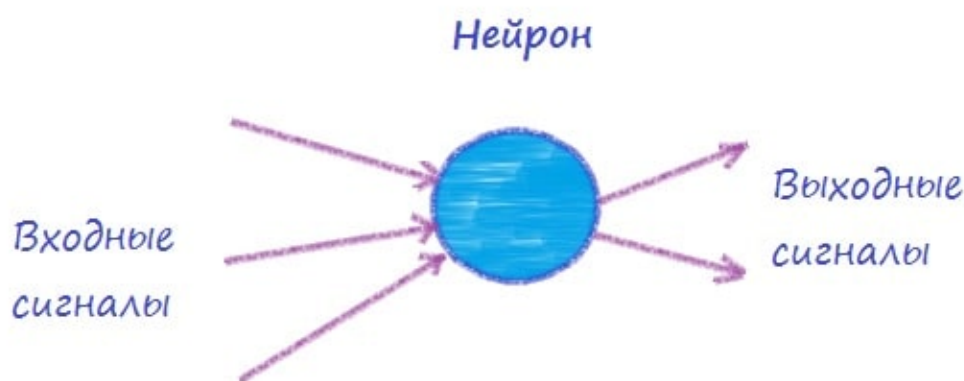
- сформировать концепцию проекта на основе потребностей участников бизнес-процесса, определить конечную цель;
- проанализировать текущие решения для реализации решения;
- сформировать конечный вид продукта (проекта) и контрольные параметры проекта;
- определить необходимые ресурсы для реализации;
- построить сеть машинного зрения;
- произвести тестирование, сравнить с контрольными параметрами.

Объектом исследования данной работы являются сами нейронные сети и принципы их работы, механизм работы систем машинного зрения, а также тестирование возможности создания собственной модели машинного зрения, или использования готовой обученной модели.

Нейронные сети – общая информация.

Изучение и использование искусственных нейронных сетей, в принципе, началось уже достаточно давно – в начале 20 века, но по-настоящему широкую известность они получили несколько позже. Связано это, в первую очередь, с тем, что стали появляться продвинутые (для того времени) вычислительные устройства, мощности которых были избыточны для работы с искусственными нейронными сетями. По сути, на данный момент можно легко смоделировать нейронную сеть средней сложности на любом персональном компьютере.

Нейронная сеть представляет из себя совокупность нейронов, соединенных друг с другом определенным образом. Рассмотрим один нейрон:



Нейрон представляет из себя элемент, который вычисляет выходной сигнал (по определенному правилу) из совокупности входных сигналов. То есть основная последовательность действий одного нейрона такая:

1. прием сигналов от предыдущих элементов сети;
2. комбинирование входных сигналов;
3. вычисление выходного сигнала;
4. передача выходного сигнала следующим элементам нейронной сети.

Между собой нейроны могут быть соединены абсолютно по-разному, это определяется структурой конкретной сети. Но суть работы нейронной сети остается всегда одной и той же. По совокупности поступающих на вход сети

сигналов на выходе формируется выходной сигнал (или несколько выходных сигналов). То есть нейронную сеть упрощенно можно представить в виде черного ящика, у которого есть входы и выходы. А внутри этого ящика сидит огромное количество нейронов.

Были перечислены основные этапы работы сети. Рассмотрим каждый из них по отдельности.

Комбинирование входных сигналов.

Поскольку к каждому нейрону могут приходить несколько входных сигналов, то при моделировании нейронной сети необходимо задать определенное правило комбинирования всех этих сигналов. В большинстве случаев используется правило суммирования взвешенных значений связей. Сейчас более подробно рассмотрим, что означает взвешенные связи.

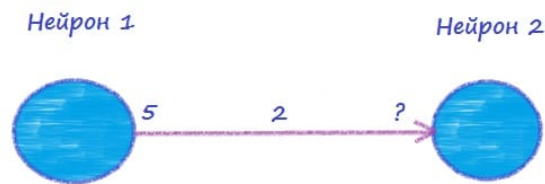
Каждую связь в сети нейронов можно полностью охарактеризовать при помощи трех факторов:

первый – элемент, от которого исходит связь;

второй – элемент, к которому связь направлена;

третий – вес связи.

Сейчас нас в большей степени интересует именно третий фактор. Вес связи определяет, будет ли усилен или ослаблен сигнал, передаваемый по данной связи. Если объяснять просто, “на пальцах”, то давайте рассмотрим такой пример:

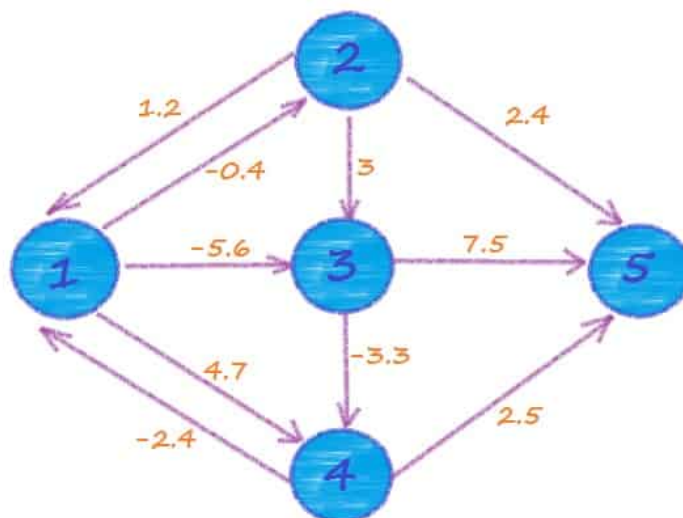


Выходной сигнал нейрона 1 равен 5. Вес связи между нейронами равен 2. Таким образом, чтобы определить входной сигнал нейрона 2, приходящий от нейрона 1, необходимо умножить значение этого сигнала на вес связи ($5 \cdot 2$).

А если сигналов много, то они все суммируются. Суммируя все взвешенные входные сигналы, мы получаем комбинированный ввод элемента сети.

Чаще всего структура связей между нейронами представляется в виде матрицы W , которую называют весовой матрицей. Элемент матрицы w_{ij} , как и в формуле, определяет вес связи, идущей от элемента i к элементу j . Для того, чтобы понять, как составляются весовые матрицы, давайте рассмотрим простую нейронную сеть:

Пример нейронной сети



Весовая матрица такой нейронной сети будет иметь следующий вид:

$$W = \begin{bmatrix} 0 & -0.4 & -5.6 & 4.7 & 0 \\ 1.2 & 0 & 3 & 0 & 2.4 \\ 0 & 0 & 0 & -3.3 & 7.5 \\ -2.4 & 0 & 0 & 0 & 2.5 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Например, от второго элемента к третьему идет связь, вес которой равен 3. Смотрим на матрицу, вторая строка, третий столбец – число 3, все верно.

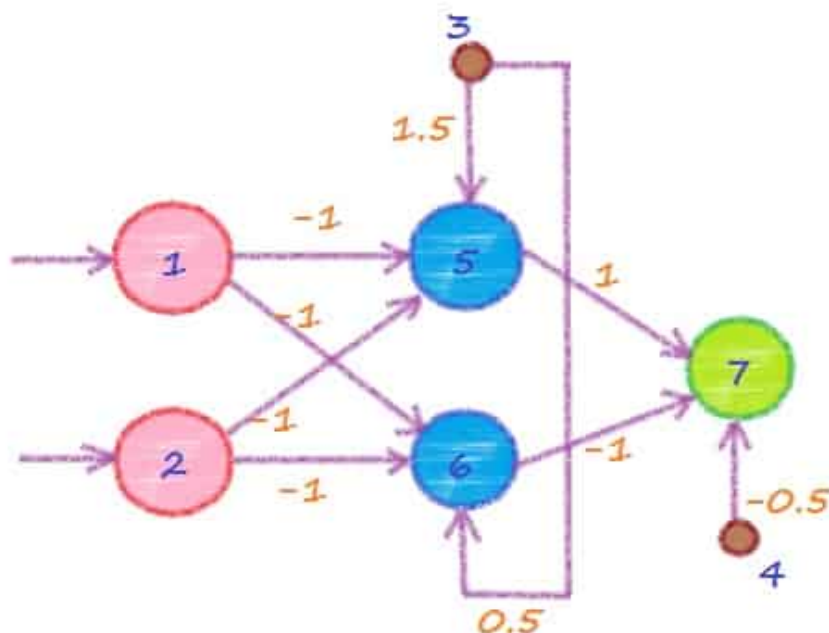
Функция активации элемента.

Рассмотрим выходные сигналы. Для каждого элемента сети имеется определенное правило, в соответствии с которым из значения комбинированного ввода элемента вычисляется его выходное значение. Это правило называется функцией активации. А само выходное значение называется активностью нейрона. В роли функций активации могут выступать абсолютно любые математические уравнения. В качестве примере рассмотрим самые популярные и часто используемые функции:

- пороговая функция – если значение комбинированного ввода ниже определенного значения (порога), то активность равна нулю, если выше – единице;
- логистическая функция.

Рассмотрим еще один небольшой пример, который очень часто используется в литературе для объяснения сути работы нейронных сетей.

Задача примера заключается в том, чтобы при помощи нейронной сети вычислить отношение XOR. То есть на вход мы будем подавать разные варианты сигналов, а на выходе должны получить результат операции XOR для поданных на вход значений:



Элементы 1 и 2 являются входными, а элемент 7 – выходным. Нейроны 5 и 6 называются скрытыми, поскольку они не связаны с внешней средой. Таким образом, мы получили три слоя – входной, скрытый и выходной. Элементы 3 и 4 называют элементами смещения. Их выходной сигнал (активность) всегда равен 1. Для вычисления комбинированного ввода в этой сети мы будем использовать правило суммирования взвешенных связей, а в качестве функции активности будет выступать пороговая функция. Если комбинированный ввод элемента меньше 0, то активность равна 0, если ввод больше 0, то активность – 1.

Подадим на вход нейрона 1 – единицу, а на вход нейрона 2 – ноль. В этом случае на выходе мы должны получить 1 ($0 \text{ XOR } 1 = 1$). Рассчитаем выходное значение вручную для демонстрации работы сети.

Комбинированный ввод элемента 5: $\text{net}_5 = 1 * (-1) + 0 * (-1) + 1 * 1.5 = 0.5$. Активность элемента 5: 1 ($0.5 > 0$).

Комбинированный ввод элемента 6: $\text{net}_6 = 1 * (-1) + 0 * (-1) + 1 * 0.5 = -0.5$. Активность элемента 6: 0.

Комбинированный ввод элемента 7: $\text{net}_7 = 1 * (1) + 0 * (-1) + 1 * (-0.5) =$

0.5. Активность элемента 7, а в то же время и выходное значение сети равно 1. Что и требовалось доказать.

Можно попробовать использовать в качестве входных сигналов все возможные значения (0 и 0, 1 и 0, 0 и 1, 1 и 1), на выходе мы всегда будем видеть значение, соответствующее таблице истинности операции XOR.

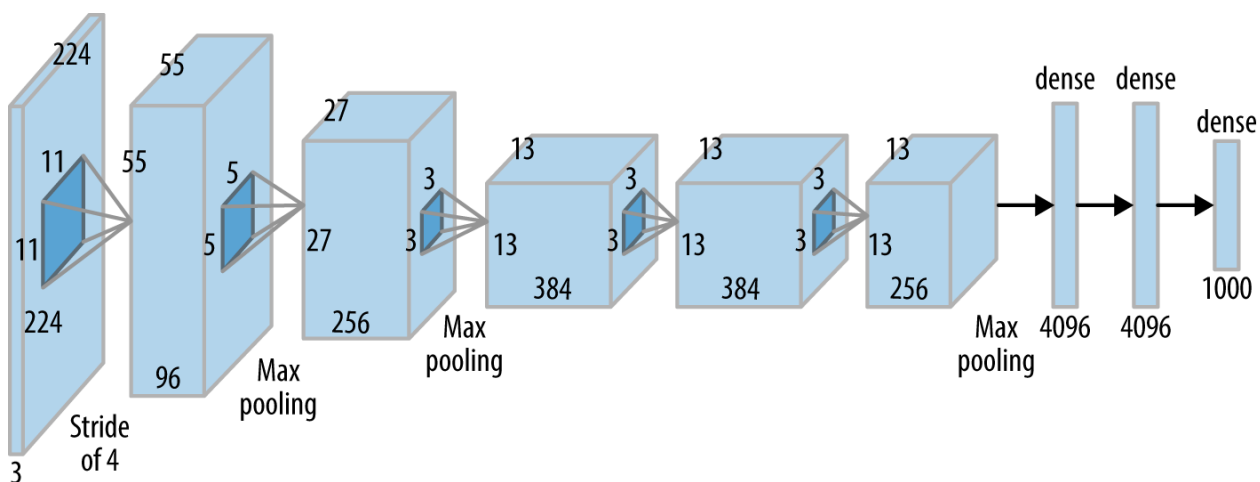
В данном случае все значения весовых коэффициентов нам были известны заранее, но главной особенностью нейронных сетей является то, что они могут сами корректировать значения веса всех связей в процессе обучения сети.

Машинное зрение – общая информация.

Машинное зрение — это научное направление в области искусственного интеллекта, в частности робототехники, и связанные с ним технологии получения изображений объектов реального мира, их обработки и использования полученных данных для решения разного рода прикладных задач без участия (полного или частичного) человека.

Сегодня существует большое количество моделей для работы с компьютерным зрением. Рассмотрим, каждую модель от первых сверточных моделей до современных.

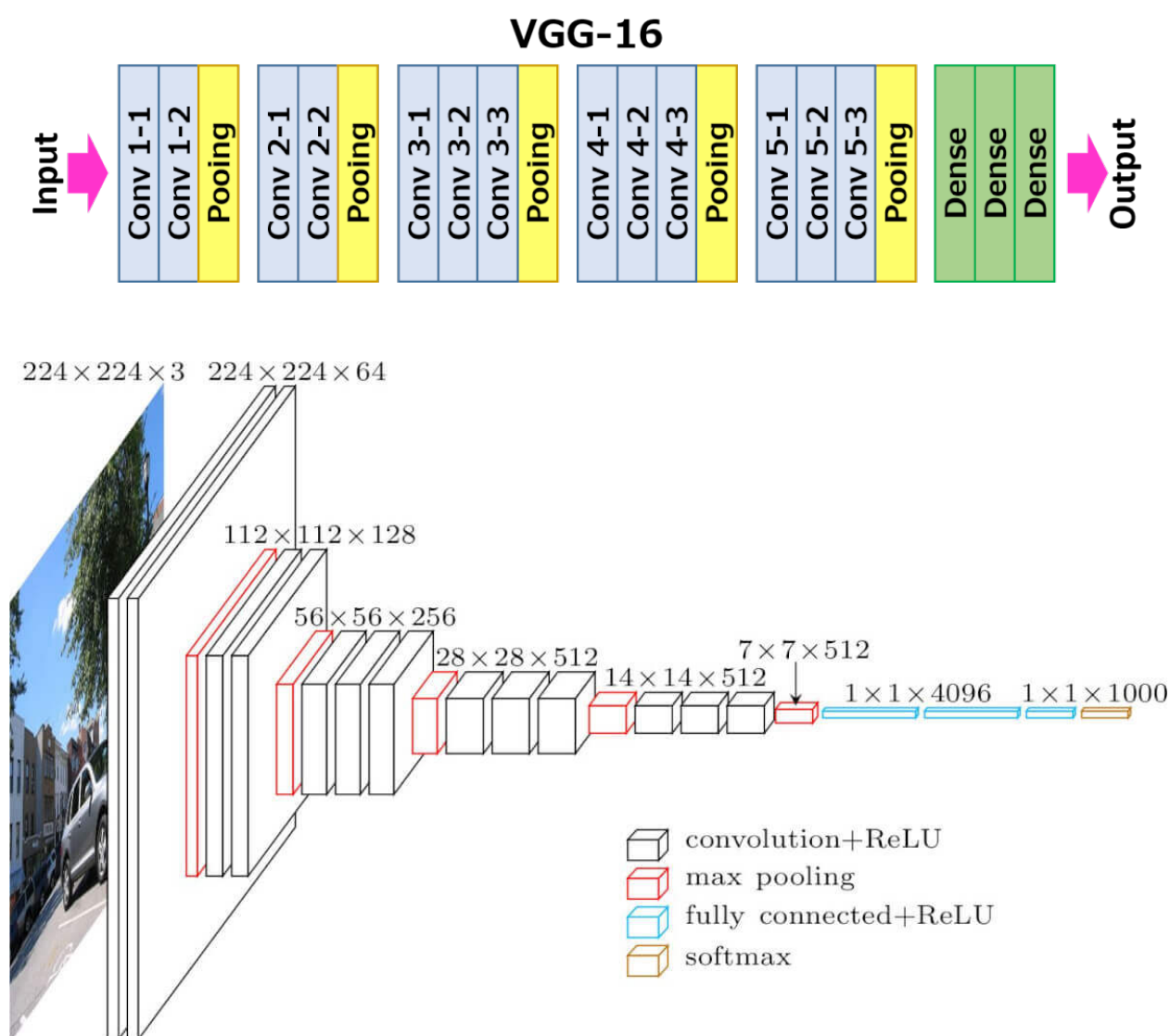
AlexNet – первая глубокая сверточная нейронная сеть. Разработчики сети выиграли конкурс по классификации изображений LSVRC-2012 на наборе данных ImageNet.



Особенности:

- принимает на вход трехканальное изображение с разрешением 224x224;
- содержит 62,3 млн. параметров;
- высокая скорость обучения за счет использования функции ReLU и по причине относительной «легкости» модели.

VGG16 – модель сверточной нейронной сети, предложенная К. Simonyan и А. Zisserman из Оксфордского университета в статье «Very Deep Convolutional Networks for Large-Scale Image Recognition». Модель достигает точности 92.7% – топ-5, при тестировании на ImageNet в задаче распознавания объектов на изображении. Этот датасет состоит из более чем 14 миллионов изображений, принадлежащих к 1000 классам. Данная модель является развитием модели AlexNet. Ее основные отличия заключаются в изменении первого и второго сверточного слоя с больших размеров слоев до размера 3x3.

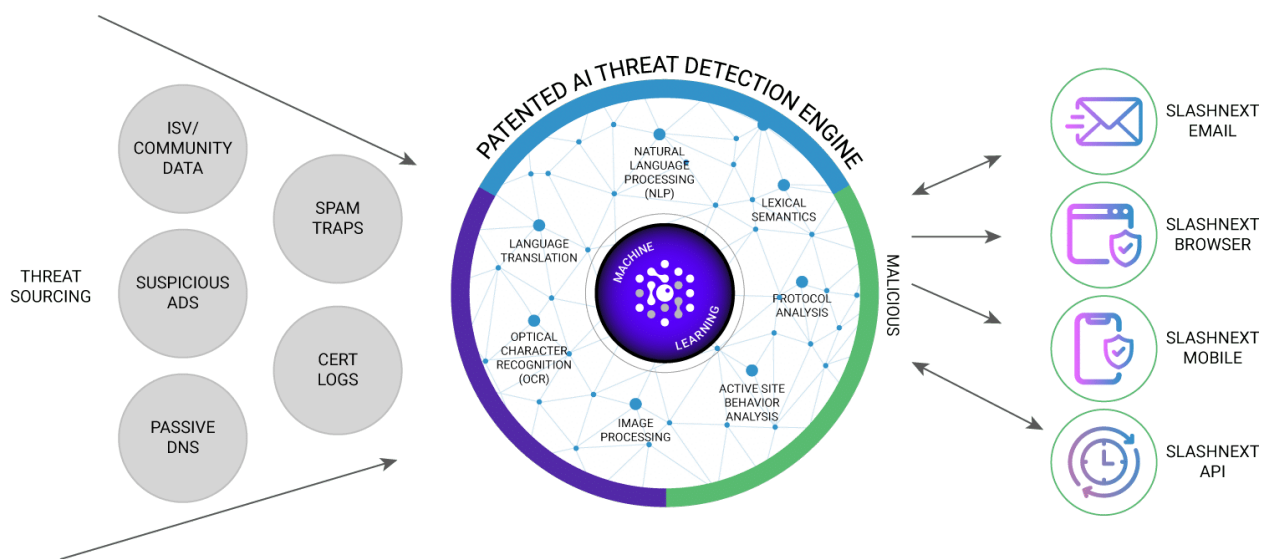


Особенности:

- все скрытые слои снабжены ReLU;

- медленная скорость обучения;
- из-за глубины и количества полносвязных узлов архитектура сети имеет большой вес (проблемы с чтением данных с диска и пропускной способностью).

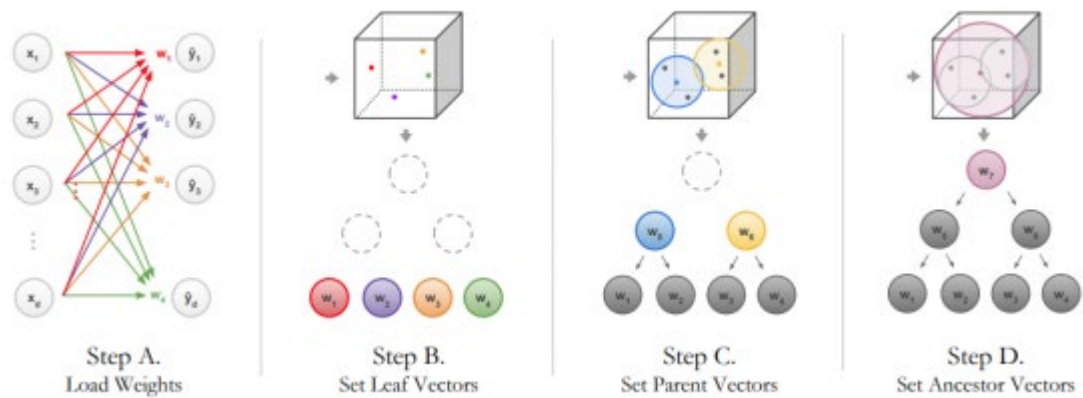
SEER – это самообучающаяся нейросеть с миллиардом параметров от FAIR для задач компьютерного зрения. Модель обучена на снимках из социальной сети Instagram, в дальнейшем модель можно дообучать на своих задачах.



Особенности:

- объединяет в себе архитектуру RegNet и формат онлайн самостоятельного обучения;
- в процессе обучения позволяет обходить ограничения по времени и задействованной памяти;
- максимальная точность предсказаний нейросети составила 84,2% на датасете ImageNet.

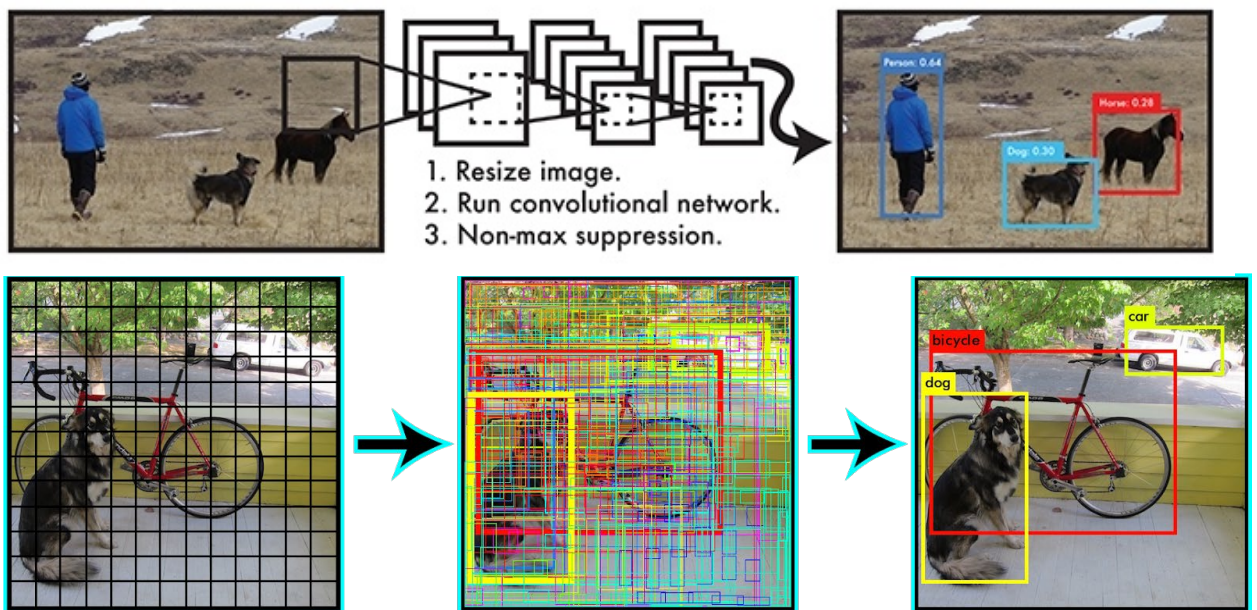
NBDT (Neural-Backed Decision Trees, NBDTs) – это нейросетевая архитектура деревьев решений для задач классификации.



Особенности:

- объединяет в себе интерпретируемость классического алгоритма дерева решений с качеством предсказаний современных нейросетей;
- модель выдает последовательные решения, которые привели к предсказанию.

YOLO (You Only Look Once) – это семейство моделей, которые стали популярны благодаря легковесности и качеству предсказаний.

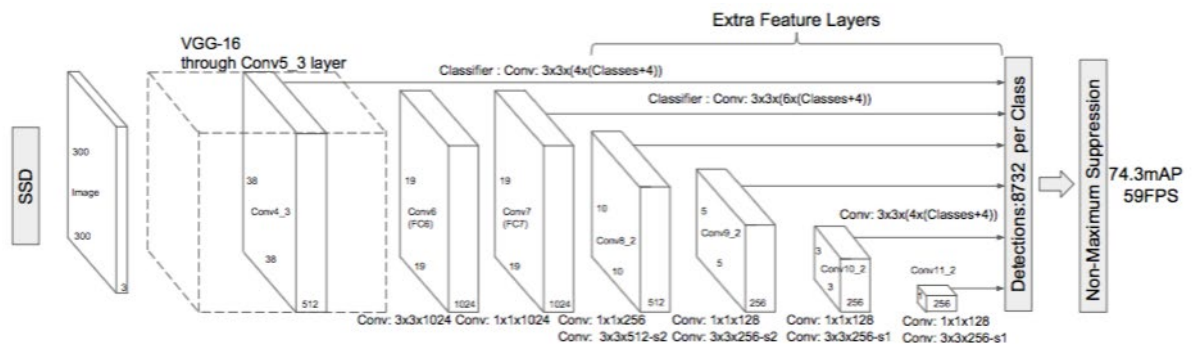


Особенности:

- алгоритм модели применяет нейронную сеть ко всему изображению, разделяя изображение на сетку и запечатлевает районы с обнаруженными элементами (якорная система);

- модель версии 4 достигает точности 55,8%;
- возможность работы на портативных устройствах и в реальном времени.

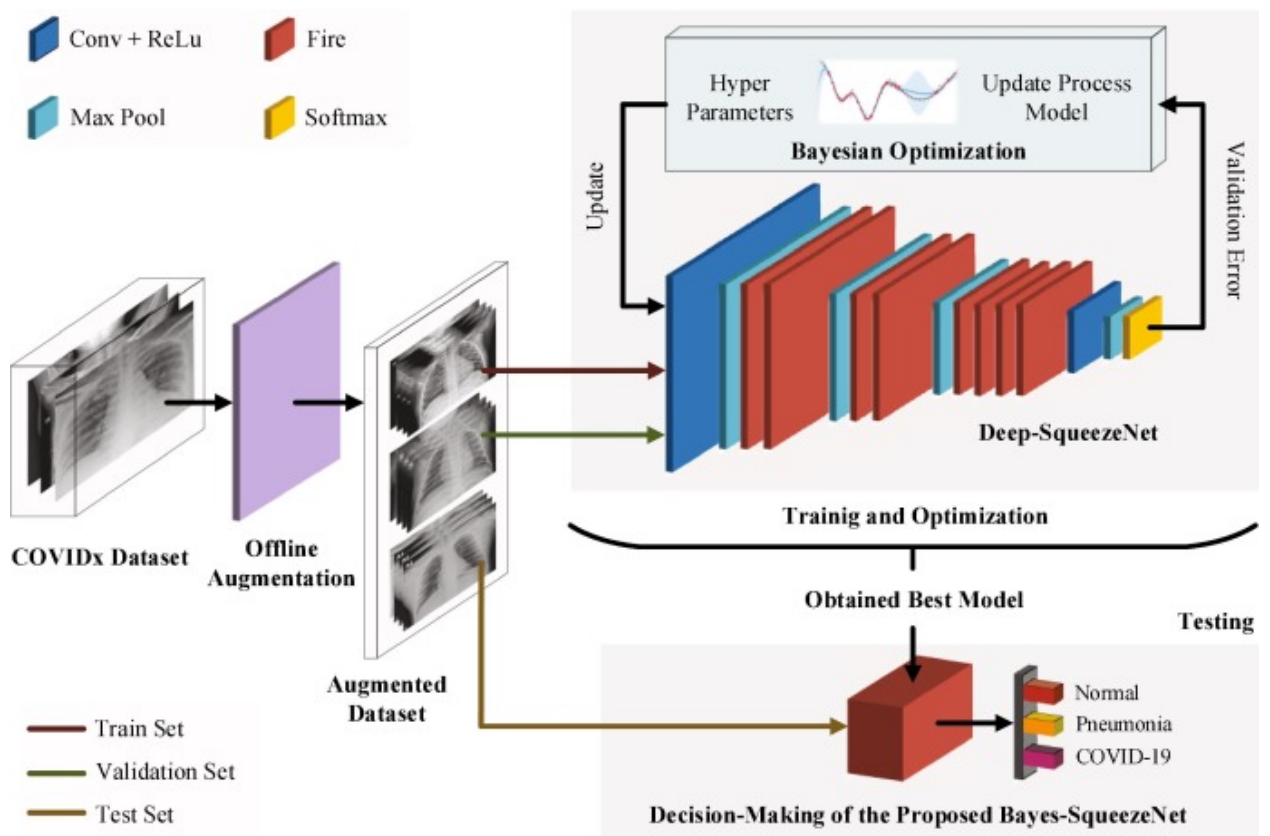
SSD (Single Shot MultiBox Detector) – сеть в которой используются наиболее удачные наработки архитектуры YOLO со своими особенностями для повышения производительности.



Особенности:

- различение объектов за один цикл работы с помощью заданной сетки окон (default box) на пирамиде изображений;
- быстрая работа;
- возможность определения за цикл как больших объектов, так и маленьких объектов.

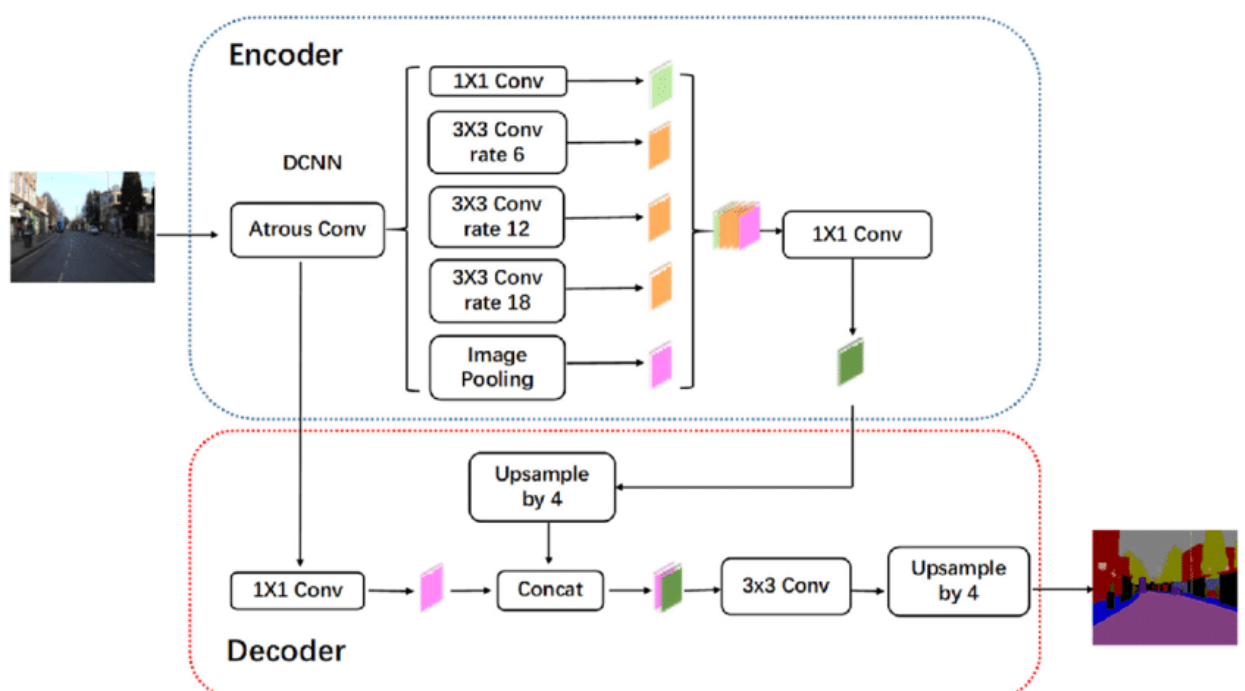
SqueezeNet – архитектура сверточной нейронной сети, в которой впервые использовался модуль Fire; модуль Fire – это модуль, который состоит из слоя сжатия и расширения, что уменьшает количество параметров и увеличивает скорость работы нейронной сети.



Особенности:

- возможность использования на мобильных устройствах;
- применяется при комбинации различных архитектур.

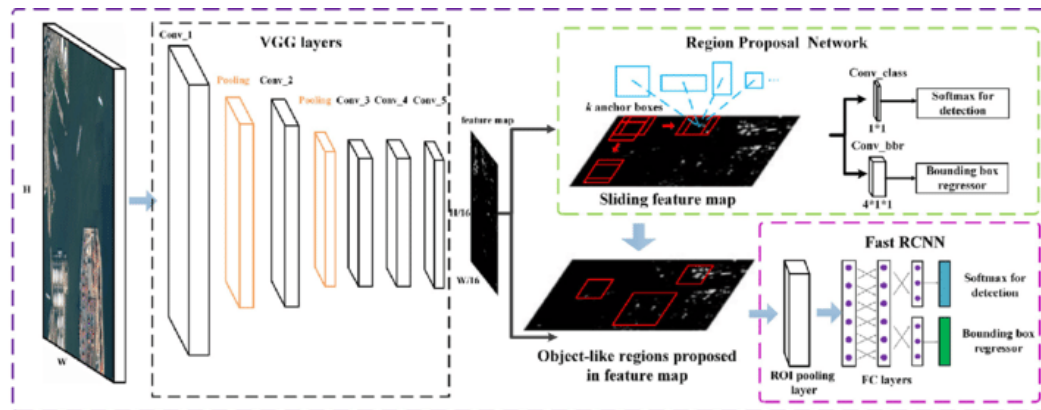
DeepLab (Semantic Image Segmentation with Deep Convolutional Nets) – модель сегментация объектов на изображении.



Особенности:

- наличие разреженной свертки (dilated convolution), которая сохраняет пространственное разрешение;
- убирает шумы в изображении и улучшает качество сегментированного объекта.

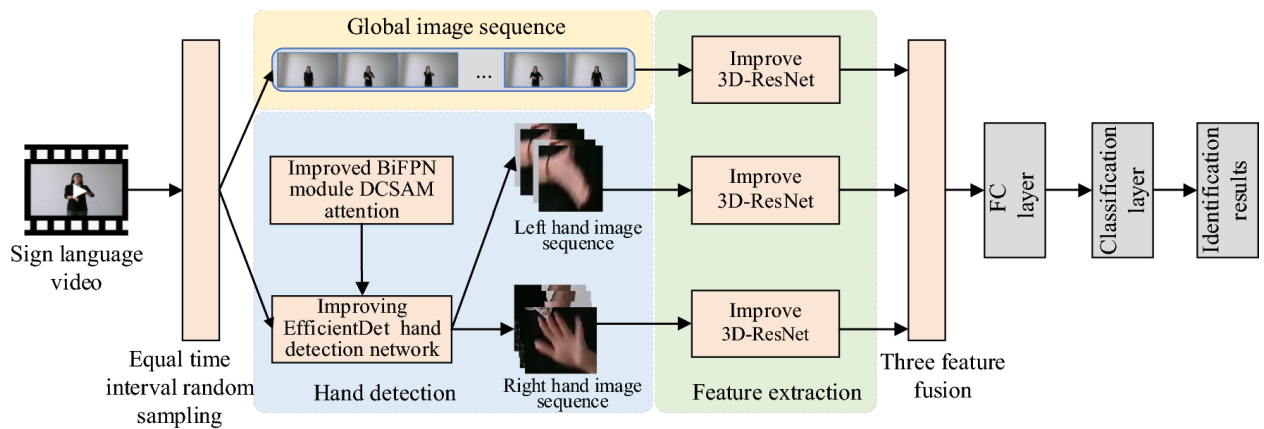
Faster R-CNN – сеть разработанная на основе R-CNN и следующей ее версии Fast R-CNN. В модели для локализации объекта вместо избирательного поиска используется Region Proposal Networks (генерация предлагаемых регионов нахождения объектов на основе последней сверточной карты признаков).



Особенности:

- модель работает быстрее Fast R-CNN, но производит локализацию объектов менее качественно;
- существуют предобученные модели.

ResNet (Residual Network, «остаточная сеть») – модель глубокой нейронной сети для классификации изображений, разработанная Microsoft для борьбы с проблемой снижения точности предсказаний с увеличением количества слоев в нейронных сетях.

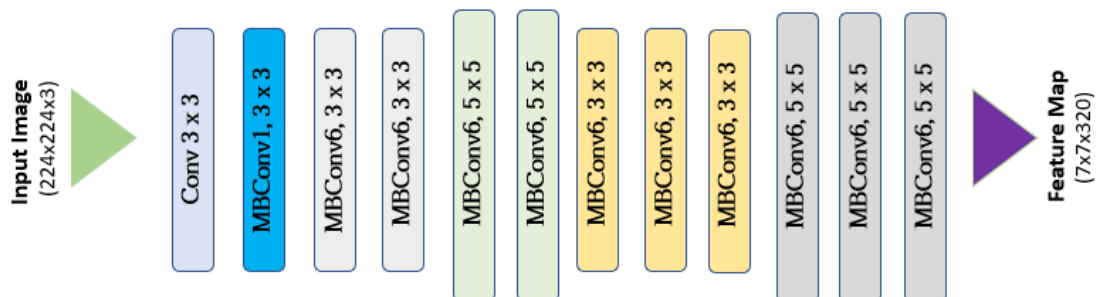


Особенности:

- после серии свёрточных блоков следуют несколько полносвязных слоёв;
- остаточное обучение сети;
- позволяет относительно легко увеличить точность благодаря увеличению глубины, чего с другими сетями добиться сложнее;
- модель можно рассматривать как несколько ансамблей параллельных или последовательных модулей.

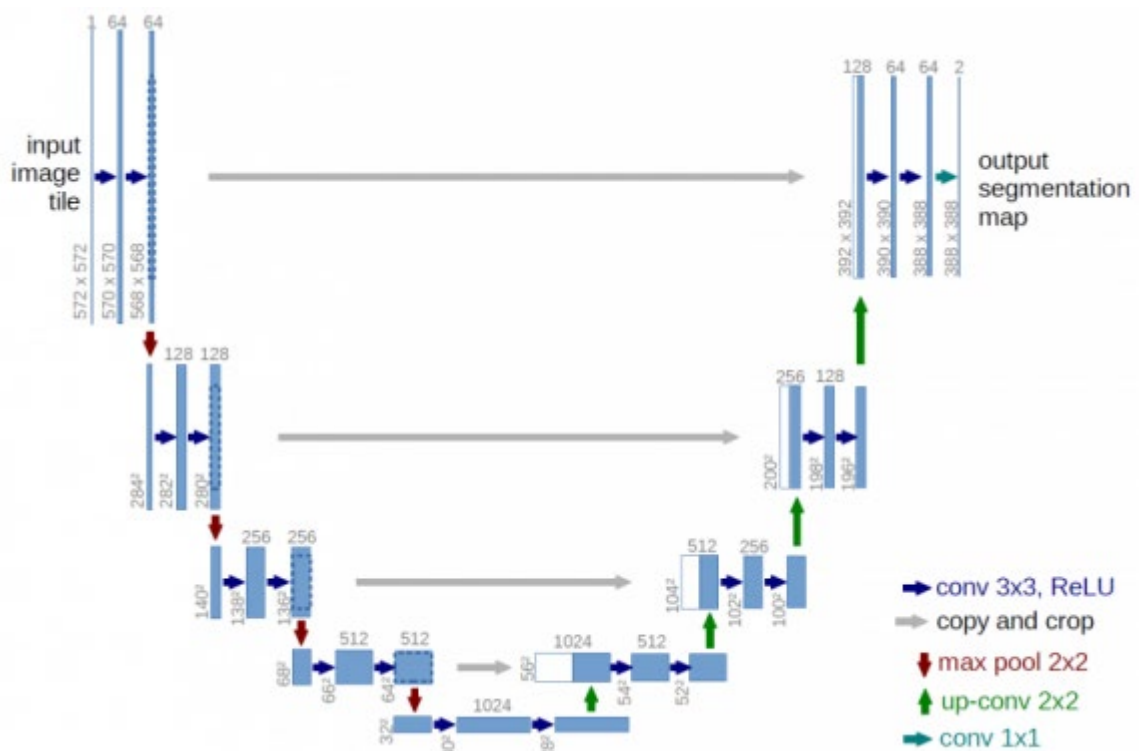
EfficientNets – класс моделей, цель разработки которых сохранить высокое качество решения задачи и повысить эффективность модели (уменьшить количество параметров и снизить вычислительную сложность).

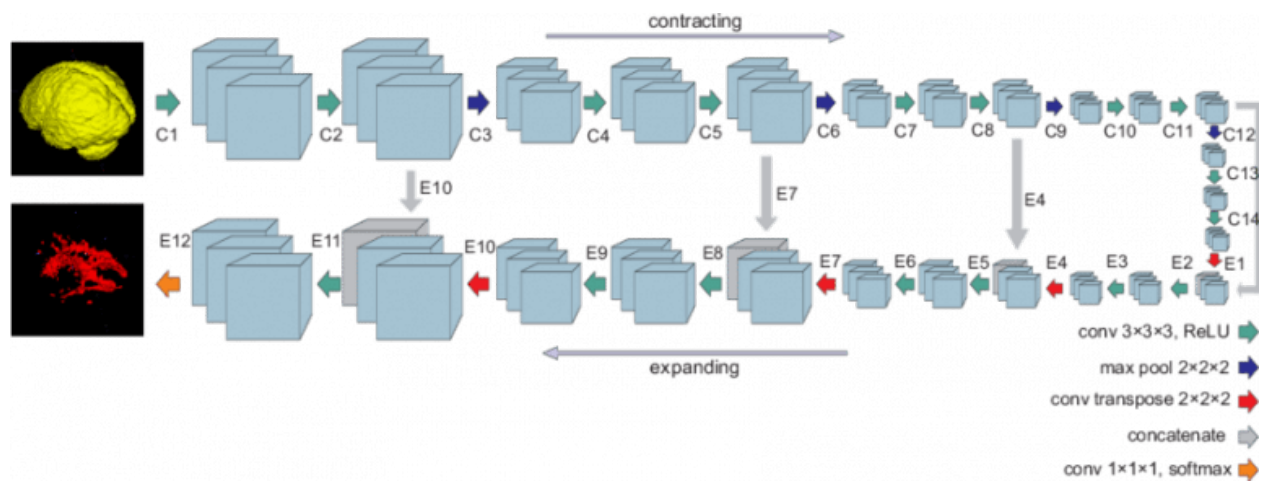
EfficientNet Architecture



- возможность самостоятельно конструирования сети (глубина, разрешение и ширина);
- высокой точности и эффективности при подборе оптимальных параметров;
- существует несколько версий модели: базовая - без преднастроенных параметров и несколько версий с уже определенными весами для параметров для возможного поиска оптимально.

U-Net - считается одной из стандартных архитектур сверточных нейронных сетей для задач сегментации изображений. Используется, когда необходимо определить класс изображения целиком и сегментировать его области по классу, т. е. создать маску, которая будет разделять изображение на несколько классов.





Особенности:

- сегментация изображения 512×512 занимает менее секунды на современном графическом процессоре;
- достижение высоких результатов в различных реальных задачах, особенно для биомедицинских приложений;
- использование небольшого количества данных для достижения хороших результатов.

Описание проблемы.

На сегодняшний день государство стимулирует и предлагает множество инструментов для развития деятельности самозанятых, а также для малого и среднего бизнеса. Но порой простых инструментов недостаточно.

Сегодня, чтобы выбрать оптимальную первую или стартовую точку даже для самого простого со стороны обывателя бизнеса, например, как общепит, нужно проанализировать большой объем информации: количество потенциальных клиентов района, количество прямых и скрытых конкурентов, наличие поставщиков и прочие.

Однако самым последним камнем преткновения становится выбор оптимального помещения для организации своего заведения. Помимо технических характеристик, в первую очередь, рассматривается один из главных факторов – пешеходный трафик. Зачем собственнику бизнеса арендовать дешевое помещение с нулевой проходимостью, либо дорогое, поток которого для него излишен. А таких помещений огромное количество, по статистке Росстата только за 2022 год в России было введено в действие 33 491 тыс. кв. м. нежилых помещений. Основным источником статистики по пешеходному потоку на сегодня является отчеты риелторов и мнение людей, живущих в непосредственной близости к помещению. Для получения более развернутой и подробной статистики необходим другой подход, который предполагает автоматизацию сбора информации.

Возможные решения, плюсы-минусы.

На сегодняшний момент бизнес исходя из запросов клиентов предлагает несколько вариантов решения данной проблемы.

Первый — это статистка из поисковиков. При запросе в поисковых сервисах, таких как Google и Yandex, выдаются релевантные результаты, но при разрешении на использовании геолокации образуется статистка, она не особо релевантная, так как запрос может быть сделан для поиска другой или аналогичной информации, сделан заранее, сделан для другого человека и прочее.

Альтернативный, но похожий вариант — это статистка из картографических сервисов. Он имеет аналогичные проблемы, но имеет один значительный плюс, эти сервисы более интегрированы в окружение пользователя. Эти сервисы собирают больше данных о клиенте и могут запрашивать информацию у пользователя ненавязчивым образом.

Следующий вариант — это сбор статистики из пешеходных картографических сервисов. Эти сервисы могут предложить большой массив данных для анализа и сбора статистики, начиная от того, где, согласно навигатору, проходило сколько человек и в какое приблизительное время, дошел ли пользователь до назначенной цели. Но у данного способа также имеются минусы: сбор и анализ информации требует значительных вычислительных мощностей; в большинстве случаев пользователи запоминают свой маршрут до конечной точки и далее следуют по заранее проложенному направлению самостоятельно без использования сервиса. Самый же значительный минус — это ограниченность в доступности данной информации. Похожие приложения имеют лишь крупные компании, и, как правило, информацией о своих пользователях они делятся неохотно или считают эту информацию служебной и не предоставляют ее третьим лицам. Во-первых, это необходимо для совершенствования собственных сервисов, приложений и бизнеса, во-вторых, для хранения такого столь массивной базы

данных нужны большие ресурсы и для упрощения хранения данной информация она преобразуется до точки отправления и конечной точки назначения, без промежуточного пути следования. Последняя причина банальная, таким образом организуется тотальная слежка за пользователем, что с точки зрения законодательства и этики запрещено.

Согласно проведенному исследованию по поисковым сервисам и специализированным сайтам на текущий момент из более современных вариантов существует два типа систем.

Первые — это подсчет трафика с помощью своего рода Wi-Fi радаров, которые в определенном небольшом радиусе регистрирует и хранит статистику обо всех устройствах с возможностью подключения к его сети. У данного метода существует свои плюсы и минусы. К плюсам можно отнести автономность работы радара, для работы системы требуется небольшие ресурсы и мощности и возможность контроля работы дистанционно. С другой стороны у данной системы малый радиус работы, она не учитывает клиентов без современных гаджетов или с выключенным протоколом Wi-Fi, необходимо специфичное оборудование и его настройка, дублирования данных в случае если у потенциального клиента имеется несколько устройств (планшет или несколько смартфонов), создание возможных помех при работе других сетей Wi-Fi и проблемы с точки зрения законодательства и этики.

Вторые — это системы подсчета клиентов на кассах магазинов. Данные системы работают на основе искусственного интеллекта и машинного зрения. В целом данная система нивелирует минусы предыдущей системы на основе подсчета пользователей Wi-Fi, такие как проблемы с точки зрения этики и законодательства, так как эти камеры расположены внутри коммерческих помещений, что является уже частной собственностью. Подсчет ведется именно по количеству людей, вошедших в помещение или совершивших покупку, а не гаджетов или потенциальных сигналов Wi-Fi. Для работы данного решения не требуется установка специального оборудования, на

сегодняшний момент практически все магазины оснащены видеокамерами или камерами безопасности. К сожалению, данная концепция не лишена минусов. Первый из них - для данной системы необходимы ресурсы в первую очередь для хранения видеоматериалов, из-за в большинстве случаев круглосуточной записи материалов и с нескольких источников. Вторым минусом — это необходимость постоянно совершенствовать и проводить повторное дообучение системы искусственного интеллекта, в первую очередь из-за различного расположения камер и различных ракурсов. Следующий негативный момент общий для всех систем искусственного интеллекта и машинного зрения с анализом видеозаписей в большом объеме связан с необходимостью большого количества вычислительных мощностей, на текущий момент для обработки одной видеозаписи требуется значительное количество вычислительных мощностей, так как видеозаписи для анализа будут поступать скорее всего с нескольких источников в связи с необходимостью полного покрытия площади, будет необходимо значительное количество ресурсов для их обработки.

Хотелось бы упомянуть способы, которые по моему мнению менее эффективны, чем упомянутые выше:

- датчики движения или пересечения определенной точки. Плюсы: стабильная работа всей системы, в том числе при ведении статистики; низкая стоимость установки и обслуживания, оборудование доступно повсеместно. Минусы: имеют узкую сферу и место использования и отсутствует возможность установки в общественных местах.

- построение модели прогнозирования на основании уже имеющихся данных о местоположении и близлежащих местах посредством искусственного интеллекта. Плюсы: нет необходимости установки и сбора данных с внешних датчиков. Минусы: неточность модели; необходимость корректировки модели по внешним факторам; постоянный сбор информации

при передислокации; высокая стоимость эксплуатации и конечной стоимости для потребителя.

- сбор информации по данным сотовых операторов. Плюсы: практически 100% покрытие информацией о всех потенциальных точках; большое количество данных и в дальней перспективе возможность создания оптимальной модели; возможность работы дистанционно, получение дополнительных сведений о возрасте и гендере. Минусы: данные о клиентах находятся в собственности операторов, как следствие высокая стоимость доступа; данные являются приблизительными; искажение данных другими устройствами или проходящим трафиком.

Данные решения могут быть использованы с учетом специфики определенного запроса клиента, исходя из местоположения, например из-за отдаленности местоположения.

На основании анализа всех изложенных фактов была предложена идея с рабочим названием «Система машинного зрения для отслеживания пешеходного трафика» с несколькими отличительными особенностями для нивелирования минусов предыдущих вариантов. Первая - камеры для данной системы будут устанавливаться в заранее определенных точках, для получения одинакового угла съемки, освещения и прочего. Это позволит анализировать записи, сделанные в идентичных условиях, что повлечет за собой повышение точности системы и даст возможность значительно увеличить тайминги дообучения системы. Следующая особенность — это выбор оптимальной точки для подсчета пешеходного трафика. Например, не непосредственно возле самого коммерческого помещения, а на рядом находящейся остановке общественного транспорта. Третья — это выбор определенного оборудования с идентичными характеристиками видеозаписей, таких как разрешение, формат, кодек, соотношение сторон и прочее. Такое решение позволит снизить повысить эффективность модели и

снизит нагрузку оборудование для обработки данных, по причине выбора единого формата видеозаписей.

Выбор модели.

После определения метода и подхода для реализации поставленной задачи следует определиться каким способом мы будем реализовывать систему. Всегда существует два пути решения любой задачи с искусственным интеллектом, первый – это создание собственной модели с нуля. Для поставленной задачи этот способ не подходит по ряду причин:

- необходимо большое количество времени на разработку модели, ее опробовании и оптимизации;
- выбор необходимого датасета для обучения, и, как следствие, время использование вычислительных ресурсов и времени для обучения;
- существование альтернативного варианта с уже обученными моделями для выполнения поставленной задачи.

В случае, если бы для выполнения задачи требовались бы специфичные требования, такие как работа под определенным оборудованием, необходимость встраивания в существующее программное обеспечение и другие условия, мы бы использовали данный способ.

Ввиду того, что ресурсы и время для реализации поставленной задачи ограничены, и необходимый смежный функционал уже частично реализован другими разработчиками, будет использован альтернативный способ, реализация проекта посредством предобученной модели.

Рассмотрим ранее перечисленные модели машинного зрения. Такие модели как AlexNet, VGG16 все еще используются для различных задач, но имеют несколько недостатков для решения нашей задачи, такие как ориентированность на работе со статичными картинками, устаревшая архитектура и тяжеловесность, и, как следствие, медленная работа.

Если же говорить о более современных моделях, таких как Overfeat, RCNN или RPN и модификаций данных моделей, данные модели используют

двухфакторные методы обнаружения объектов. Сперва идет поиск «подозрительных» зон на изображении, зон, возможно, содержащих интересующие нас объекты. Второй этап заключается в классификации зон, найденных на первом этапе. Данные модели имеют высокую точность обнаружения объектов с относительно невысокой скоростью работы.

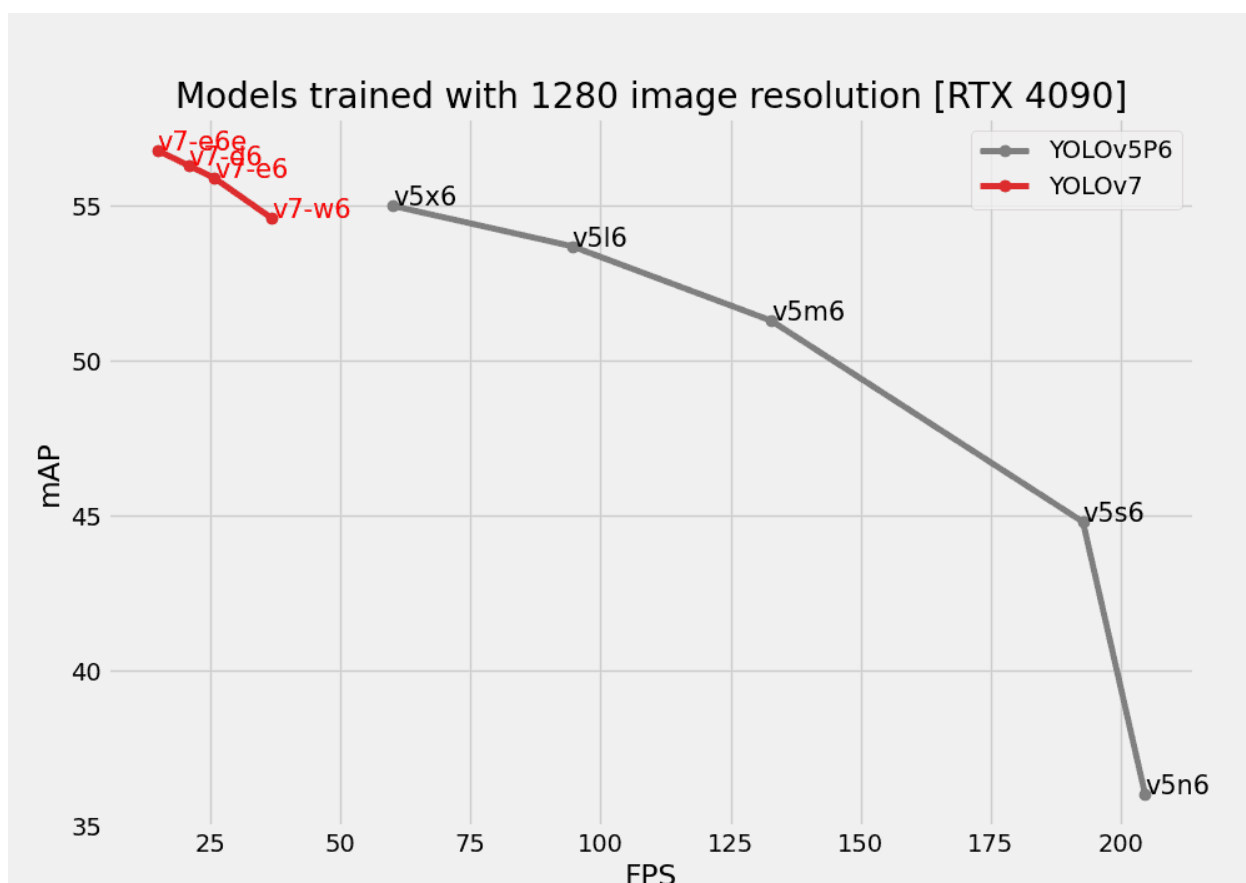
Несмотря на это для решения нашей задачи оптимальным выбором будут модели с One-stage подходом. Одноэтапные нейросетевые алгоритмы детектирования объектов на изображении не включают в себя стадию поиска подозрительных зон на изображении, а сразу нацелены на обнаружение объектов. Преимуществами данного вида алгоритмов являются их простота и относительно высокая скорость работы, а среди их недостатков можно отметить более низкую точность детектирования объектов по сравнению с двухэтапными алгоритмами, а также меньшую гибкость алгоритма. Наиболее распространенными одноэтапными моделями являются YOLO и SSD.

Для определения оптимальной модели было произведено сравнение моделей Faster R-CNN, SSD и YOLOv3. По результатам наивысший процент mAP (mean average precision) получила модель Faster R-CNN (87,7%), однако скорость обработки оказалась наихудшая из представленных моделей и недостаточная для анализа видеопотока в реальном времени. Точность модели SSD оказалась немного хуже по сравнению с Faster R-CNN, показатель mAP составил 82,4%, при этом скорость обработки данных увеличилась на 360%. Модель YOLOv3 показала наибольший показатель по скорости обработки данных, рост по сравнению с SSD составил 63%, однако точность модели оказалась наихудшая из всех исследованных, показатель mAP составил 80,2%.

На начальном этапе реализации проекта рассматривалась обработка данных видео в записи, поэтому оптимальной с точки зрения точности и затрачиваемых ресурсов является модель SSD. Однако в дальнейшем, в случае расширения проекта, необходимости получения данных централизованно и автоматически, с большого количества источников и в режиме реального

времени необходимо выбрать модель YOLO, так как при небольшом падении точности она имеет наибольшую производительность и как следствие эффективность.

Сегодня модель YOLO имеет множество версий от 3 версии, выпущенной в 2018 году до 8 версии, выпущенной в 2023 году. В качестве испытательного стенда была взята сборка с самым мощным графическим процессором NVIDIA GeForce RTX 4090 с поддержкой CUDA. В качестве тестового материала брались изображения с разрешением 1280 как наиболее распространенное разрешения для видеокамер, устанавливаемых повсеместно. Для сравнения брались показатели mAP и скорость обработки данных (FPS). Из результатов видно, что чем меньше модель, тем она быстрее обрабатывает данные, при этом качество модели также падает. В целом модели семейства YOLOv5 показали себя с лучшей стороны в соотношении качества работы модели и ее производительности.



Однако в данном тесте не участвовала модель YOLOv8. Основные преимущества 8 версии по сравнению с предыдущими:

- данная модель быстрее и точнее предыдущих сборок;
- новая основная (backbone) сеть служащая для извлечения первоначальных признаков и поступающего на вход изображения;
- новая Anchor-Free head, алгоритм построения возможных областей нахождения объектов около якорных точек;
- новая функция потерь.

Данная категория модели аналогично содержит пять моделей YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l и YOLOv8x, модели расположены от самой быстрой и маленькой к самой точной, но медленной.

Далее было произведено сравнение моделей YOLOv5, как наиболее оптимальной по предыдущему тесту, и YOLOv8 по разным моделям. Так как изначально модель YOLOv8 предобучена на работе с изображениями с разрешением 224 и 640 сравнение будет по картинкам в данном разрешении.

Model Size	Detection*	Segmentation*	Classification*
Nano	+33.21%	+32.97%	+3.10%
Small	+20.05%	+18.62%	+1.12%
Medium	+10.57%	+10.89%	+0.66%
Large	+7.96%	+6.73%	0.00%
Xtra Large	+6.31%	+5.33%	-0.76%

*Image Size = 640

*Image Size = 224

Сравнение производительность моделей YOLOv8 к моделям YOLOv5

По результатам более современные модели YOLOv8 показывают себя лучше, их точность и производительность лучше моделей YOLOv5, за исключением одной из классификационных моделей.

Однако модель YOLOv5 обладает следующими преимуществами по сравнению с более современной моделью:

- большая база мануалов и инструкций, так как комьюнити уже использует данную модель более продолжительное время и в больших нейронных сетях;
- с момента релиза в 2020 году, данная модель уже получила множество исправлений ошибок и багов, как следствие модель более стабильна;
- в случае возникновения ошибок или багов имеется огромное количество информации в открытых источниках.

Несмотря на эти преимущества модели YOLOv5 для реализации проекта будет использоваться модель YOLOv8 как более быстрая и современная, с момента ее релиза уже прошло достаточно времени, были исправлены множество ошибок и сообщество создало обширную базу данных. В случае возникновения критических ошибок, которые невозможно исправить, совершить замену модели можно будет в кратчайшие сроки.

В качестве детектора объектов (object detection) будет использован YOLOv8, то в качестве трекера объектов (object tracker) будет использоваться DeepSORT по следующим причинам:

- возможность использовать в связке с YOLO для работы в режиме реального времени, так как решение является легковесным;
- простая реализация и инициализация с YOLO;
- наличие форумов и комьюнити, для поиска решения ошибок или нахождения мануалов;

- возможность отслеживания объекта при его перекрытии другим объектом или его потери в кадре;
- решение изначально обучено для трекинга людей в кадре.

Определение конечного результата, определения контрольных параметров успешной реализации.

После определения, как и каким инструментом мы будем пользоваться для реализации идеи, нужно определить конечный результат и контрольные параметры или критерии и их количественные показатели, которые будут сигнализировать, что промежуточные контрольные точки и конечный результат достигнут.

Цель проекта: создать систему на основе искусственного интеллекта и машинного зрения для контроля пешеходного трафика посредством видеозаписей или видео в реальном времени в автоматическом режиме с точностью не менее 30%.

Ключевые показатели:

- анализ видеофайлов в автономном режиме;
- ведение базы данных статистики;
- увеличение качества подбора коммерческой недвижимости в 2 раза.

Выбор оптимального оборудования.

Для полноценной работы проекта, даже в один полный цикл, требуется несколько составляющих:

- высокопроизводительный персональный компьютер, на данном компьютере будет производиться основная часть проекта, а контрено будут обрабатываться и анализироваться видеозаписи машинным зрением на базе искусственного интеллекта;

- сервер, необходимый для хранения видеозаписей и статистики по уже отработанным материалам. Также данный через данный компьютер будут подаваться в очередь видеозаписи на высокопроизводительный компьютер. Для реализации проекта в коммерческую составляющую, на данном компьютере будет формироваться бухгалтерская отчетность и вся другая необходимая документация;

- видеокамеры и аксессуары к ним, необходимы для съемки видеоматериала для дальнейшего анализа. Видеокамеры необходимы двух типов: для постоянного размещения на статичных общественных объектах, например антивандальные камеры для установки непосредственно возле коммерческих помещений, и камеры временного размещения, например видеорегистраторы в автомобилях или ip-видеокамеры для размещения на временных точках;

- сопутствующее оборудование: маршрутизаторы, карт-ридеры, источники бесперебойного питания, принтеры и МФУ;

- программное обеспечение, сюда входит целый сектор программ: операционные системы, офисные пакеты, программное обеспечение для бухгалтерской отчетности, антивирусы, редакторы видеофайлов и программное обеспечение для ведения базы данных и редакторы исходного кода;

- мебель, расходные материалы и канцтовары;

- аренда помещения, заработная плата сотрудников;
- первичная настройка оборудования и программного обеспечения.

Подробно пройдемся по каждому пункту. Для расчета базовой стоимости оборудования для проекта использовалась средняя цена одного из самого крупнейшего ретейлера электроники, для расчета услуг бралась средняя цена по Красноярску.

В случае реализации проекта собственными силами, расходов для реализации не включали бы в себя: аренду помещений – домашнее помещение; частично заработную плату сотрудникам – выполнение обязанностей своими силами; первичная настройка оборудования и программного обеспечения. Но для того, чтобы оценить реальную эффективность проекта, в том числе для возможной продажи проекта под ключ или реализацию проекта в виде продажи готового бизнес-плана, все возможные расходы будут включены в конечную смету.

Для работы искусственного интеллекта на сегодняшний день видеокарта является основным критерием, в том числе и для системы обработки видеозаписей. На текущий момент существует профессиональные ускорители, специализированные на работу с искусственным интеллектом, однако доступность и высокая цена профессиональных решений не являются оптимальными для проекта. Тем не менее, для реализации проекта будет достаточно рабочего стенда на базе видеокарты NVIDIA Geforce RTX 4080 и выше, процессором, соответствующим данной видеокарте, оперативной памятью размером 32 гб. и быстрым твердотельным накопителем (SSD). Примерная стоимость такого оборудования, вместе с монитором, клавиатурой и мышкой, на текущий момент составляет примерно 280 тыс. руб.

Для серверного компьютера основным критерием будет являться объем памяти жестких дисков и их скорость. В качестве оптимального оборудования брался средний сегмент процессоров со встроенным видеоускорителем, 8 гб.

оперативной памяти и материнская плата, с необходимым количеством разъемов для подключения жестких дисков. Приблизительная стоимость персонального компьютера, со всей необходимой периферией и с объемом твердотельных накопителей 10 тб. составит примерно 80 тыс. руб.

В качестве начального этапа работы проекта будет рассмотрена покупка трех видеокамеры разного типа (ip-видеокамера, регистратор и антивандальная камера), для возможности работы в разных условиях, с необходимыми комплектующими, такими как кабеля, крепление и карты памяти. Суммарная стоимость всех трех комплектов составляет 30 тыс. руб.

Для сопутствующего оборудования бралось оборудование среднего и базового уровня. Стоимость двух источника бесперебойного питания составляет 10 тыс. руб. Карт-ридер с возможностью чтения всех современных форматов карт памяти стоит 4 тыс. руб. Цена черно-белого МФУ составляет 16 тыс. руб. На все сопутствующее оборудование необходимо 30 тыс. руб.

Часть необходимого программного обеспечения распространяется бесплатно, например: операционная система на базе Linux для основного компьютера, видеоредакторы, офисные пакеты и редакторы исходного кода. Другое программного обеспечение необходимо приобретать единоразово или ежегодно продлять подписку. При оформлении подписки рассмотрим вариант с 3 годовой подпиской. Операционная система windows – 4 тыс. руб. Подписка на антивирус 5 тыс. руб. Программное обеспечение для бухгалтерской отчетности 6 тыс. руб. Стоимость всего программного обеспечения составляет 15 тыс. руб.

Для расчета стоимость мебели, расходных материалов, канцтоваров, аренду офиса и заработной платы сотрудников возьмем среднюю стоимость для малого бизнеса по стране. В расчете на первые три месяца работы данная статья расходов составит 480 тыс. руб.

Стоимость настройки и оборудования такой спецификации составит 100 тыс. руб.

Итого: стоимость всего необходимого для старта проекта составит 1 млн. 15 тыс. руб.

Первичный код.

Для начала начинаем установку необходимых библиотек:

```
Pip install tensorflow
```

```
>> pip install numpy
```

```
>> pip install os
```

```
>> pip install random
```

```
>> pip install cv2
```

```
>> pip install ultralytics
```

```
>> pip install scipy
```

Далее мы создаем в редакторе кода файл трекера `tracker.py`, который будет работать с помощью DeepSORT:

```
from scipy.optimize import linear_sum_assignment as linear_assignment
```

```
from deep_sort.deep_sort.tracker import Tracker as DeepSortTracker
```

```
from deep_sort.tools import generate_detections as gdet
```

```
from deep_sort.deep_sort import nn_matching
```

```
from deep_sort.deep_sort.detection import Detection
```

```
import numpy as np
```

```
class Tracker:
```

```
    tracker = None
```

```
    encoder = None
```

```
    tracks = None
```

```

def __init__(self):

    max_cosine_distance = 0.4

    nn_budget = None

    encoder_model_filename = 'model/mars-small128.pb'

    metric = nn_matching.NearestNeighborDistanceMetric("cosine",
max_cosine_distance, nn_budget)

    self.tracker = DeepSortTracker(metric)

    self.encoder = gdet.create_box_encoder(encoder_model_filename,
batch_size=1)

def update(self, frame, detections):

    if len(detections) == 0:

        self.tracker.predict()

        self.tracker.update([])

        self.update_tracks()

        return

    bboxes = np.asarray([d[:-1] for d in detections])

    bboxes[:, 2:] = bboxes[:, 2:] - bboxes[:, 0:2]

    scores = [d[-1] for d in detections]

```

```

features = self.encoder(frame, bboxes)

dets = []

for bbox_id, bbox in enumerate(bboxes):
    dets.append(Detection(bbox, scores[bbox_id], features[bbox_id]))

self.tracker.predict()

self.tracker.update(dets)

self.update_tracks()

def update_tracks(self):
    tracks = []

    for track in self.tracker.tracks:
        if not track.is_confirmed() or track.time_since_update > 1:
            continue

        bbox = track.to_tlbr()

        id = track.track_id

        tracks.append(Track(id, bbox))

```



```
self.tracks = tracks
```

```
class Track:
```

```
    track_id = None
```

```
    bbox = None
```

```
    def __init__(self, id, bbox):
```

```
        self.track_id = id
```

```
        self.bbox = bbox
```

Предварительно перед этим нужно скачать модель mars-small128 для работы трекера и сохранить ее предварительно в папку model, для того чтобы все работало корректно:

```
encoder_model_filename = 'model_data/mars-small128.pb'
```

Данную модель можно свободно найти в интернете.

Далее нам необходимо клонировать в наш локальный репозиторий официальную сборку DeepSORT. Для этого в интернете находим ссылку на данный репозиторий и с помощью команды клонируем:

```
git clone https://github.com/computervisioneng/deep_sort.git
```

Если у нас установлены последние версии библиотек, то нам необходимо заменить одну строчку в файлах клонированного репозитория. По пути deep_sort/deep_sort в файле linear_assignment.py необходимо заменить строчку:

```
from sklearn.utils.linear_assignment_ import linear_assignment
```

На строчку:

```
from scipy.optimize import linear_sum_assignment as linear_assignment
```

Это делается по причине разницы библиотек старой и новых версий.

Следующим шагом основного инициализатора нашей системы машинного зрения. Создаем в нашем редакторе кода файл `main.py`.

В первую очередь мы проверяем можем ли мы просматривать выбранное для анализа видео.

Далее мы приступаем к следующему этапу и добавляем код для детекции. Импортируем ранее созданный файл трекера и иницилируем его в нашем коде. После получения координат и классов объектов необходимо переопределить их в простые числа для координат. Трекер будет автоматически делать захват и определения объектов, и их обрамление в рамки, каждого по отдельности и их идентификации по классам и вероятности. Каждому объекту видеозаписи будет присваиваться уникальный `id` для корректного отслеживания.

После того как на видео происходит идентификация объектов и разметка их боксов, необходимо прописать код сохранения видео с готовым результатом. Необходимо указать кодек формирования видео, и его исходный фреймрейт.

Также необходимо добавить два параметра, при вероятности менее 0,5 не учитывать объекты на видео и учитывать только класс 0 (люди).

В конце нашего кода необходимо добавить строчку вывода уникальных `id` – это и будет количество людей, попавших в видеозапись.

```
import os
```

```
import random
```

```

import cv2

from ultralytics import YOLO

from tracker import Tracker

video_path = os.path.join('.', 'data', test.mp4')

video_out_path = os.path.join('.', 'test_out.mp4')

cap = cv2.VideoCapture(video_path)

ret, frame = cap.read()

cap_out = cv2.VideoWriter(video_out_path,
cv2.VideoWriter_fourcc(*'MP4V'), cap.get(cv2.CAP_PROP_FPS),
(frame.shape[1], frame.shape[0]))

model = YOLO("yolov8n.pt")

tracker = Tracker()

colors = [(random.randint(0, 255), random.randint(0, 255), random.randint(0,
255)) for j in range(10)]

detection_threshold = 0.5

```

```
while ret:
```

```
    results = model(frame)
```

```
    for result in results:
```

```
        detections = []
```

```
        for r in result.bboxes.data.tolist():
```

```
            x1, y1, x2, y2, score, class_id = r
```

```
            x1 = int(x1)
```

```
            x2 = int(x2)
```

```
            y1 = int(y1)
```

```
            y2 = int(y2)
```

```
            class_id = int(class_id)
```

```
            if score > detection_threshold:
```

```
                detections.append([x1, y1, x2, y2, score])
```

```
        if class_id == 0:
```

```
            tracker.update(frame, detections)
```

```
    for track in tracker.tracks:
```

```
        bbox = track.bbox
```

```
        x1, y1, x2, y2 = bbox
```

```
track_id = track.track_id

cv2.rectangle(frame, (int(x1), int(y1)), (int(x2), int(y2)),
(colors[track_id % len(colors)], 3)

cap_out.write(frame)

ret, frame = cap.read()

cap.release()

cap_out.release()

cv2.destroyAllWindows()

print(len(track_id))
```

Тестирование системы, результаты.

Возможность произвести тестирование системы на домашнем компьютере отсутствовала по причине конфликта операционной системы, драйверов и CUDA. Произвести захват фото и видео веб камеры оказалось возможным.



На рабочем стенде с тестовым видео ключевые критерии системы оказались на уровне ключевых ранее определенных показателей.

Точность определения объектов предобученной системы уже были выше целевого показателя.

По ключевым показателям:

- автоматический анализ видеозаписей удалось организовать;
- реализовать базу данных статистики по видеозаписям в условиях сжатых сроков организовать не удалось, автоматический подсчет количества людей в видеозаписи отображается в консоле;
- увеличение подбора коммерческой недвижимости в 2 раза определить невозможно, проект не запущен в коммерческую эксплуатацию.

При подсчете экономической эффективности следует учитывать несколько параметров. Помимо единоразовых расходов на покупку оборудования существуют постоянные расходы на текущую эксплуатацию (интернет, электричество и прочее) в размере 10 тыс. руб.

В отличии от геомаркетинга, стоимость нашего оборудования будет рассчитываться из расчета количества работы видеокамеры, вместе с дальнейшей статистикой. Оценочная средняя стоимость одного дня замера для помещения составит 2,5 тыс. руб. При средней загруженности оборудования 25 дней в месяц, доход составит 188 тыс. руб.

При единоразовых затратах 1 млн. 15 тыс. руб., среднемесячных расходах в размере 10 тыс. руб. и 160 тыс. на аренду и заработную плату, и среднемесячным доходом в 188 тыс. руб. средняя окупаемость проекта составит 3 года. Это является приемлемым при реализации проектов.

Заключение.

На сегодняшний день понятие «искусственный интеллект» все чаще упоминается повсюду, за последние десять лет разработчики нейронных сетей и сообщество, добились огромного прогресса в развитии и популяции данного направления. Все больше сфер с помощью искусственного интеллекта автоматизируют различный функционал, проводят аналитику и даже прогнозируют будущий результат. И данная технология применяется даже в самых неожиданных местах, например придумывает новые танцы, разрабатывает модную одежду, придумывает новые рецепты или вовсе ведет прогноз погоды. Модели, которые еще использовались несколько лет назад, не успевают за текущими моделями в качестве обработки исходных данных, скорости обучения и выдачи готового результата. Немаловажную роль в этом сыграли два фактора. Первый — это цифровизация всех возможных данных и, как следствие, появления большого массива данных для обучения, начиная с инструкций и рецептов, заканчивая переписками в социальных сетях. Второй фактор заключается в росте в геометрической прогрессии вычислительных мощностей. Для обучения и работы нейронной сети достаточно домашнего видеоускорителя средней сегмента работающего на специальных библиотеках (CUDA). Кроме того, сегодня выпускаются специальные ускорители, специализированные для работы с искусственным интеллектом, такое оборудование как NVIDIA A100 значительно превосходит любой домашний персональный компьютер в работе с ИИ.

Что касается такого направления нейронных сетей как машинное зрение, сегодня оно находится на втором месте по использованию, после генеративных систем. Системы, построенные на машинном зрении, в первую очередь, создавались для идентификации и сегментации объектов. Это автоматизировало и ускоряло множество процессов, но область применения данных систем расширилась. Данные системы применяют как составную часть генеративных систем, в автопилотах и робототехнике, и для повышения

уровня безопасности. Катализатором развития данных систем послужило внедрения новых механизмов, таких как сверточные сети, развитие оборудования для съемки, мобильные камеры, лидары и ARM процессоры с ядрами, направленными на работу с искусственным интеллектом, и доступность большого количества информации в открытом доступе для обучения сетей.

Сегодня развитие малого и среднего бизнеса для людей и государства считается драйвером развития экономики. Предприниматели, например в ритейле, для оптимизации расходов, стараются переводить свой бизнес в «виртуальное пространство», оставляя лишь пункты выдачи товаров или предоставления конечной услуги. Исходя из опроса действующих предпринимателей и поиска информации в интернете, довольно остро стоит проблема оценки интенсивности пешеходного трафика около определенных точек — это один из ключевых факторов в формировании стратегии бизнеса и стоимости аренды/покупки коммерческого помещения. Эта проблема и стала основой для будущего проекта. В ходе анализа существующих решений, которые применяются по всему миру, было сформировано оптимальное решение для проекта. Для проекта под названием «Система машинного зрения для отслеживания пешеходного трафика» была проанализирована текущая ситуация, сформирована цель и ключевые показатели, определена необходимая архитектура для реализации с нуля.

Проект был реализован посредством моделей YOLO и DeepSort. В процессе разработки и настройки нейронной сети были использованы навыки, полученные в ходе обучения по курсу, в том числе для решения проблемных ситуаций. Итог реализации проекта:

1. цель проекта достигнута, но с некоторыми нюансами. разметка и подсчет людей идет в автоматическом режиме из заранее загруженной видеозаписи с точностью 30% и более. Отсутствует реализация удобного и

понятного интерфейса для загрузки видеофайлов в нейросеть. Функционал разметки видеопотока в реальном времени отсутствует;

2. режим автономной работы с серверной машины не реализован;
3. в пилотном запуске данные по пешеходному потоку отображаются в режиме консоли, база данных не ведется.

Для запуска проекта в коммерческое использование требуется доработка ряда функционала и разработка графического интерфейса для комфортной работы.

Была просчитана экономическая эффективность проекта. При средней загрузке оборудования, собственной разработки и настройки системы окупаемость проекта наступит через 3 года. Проект экономический целесообразен. При расширении базы видеорегистраторов окупаемость проекта сократится, так как самым затратным является оборудование для хранения и обработки видеофайлов.

В целом при планировании данного проекта продумывались и последующие этапы в разработке системы машинного зрения. Создание централизованной базы данных, в которой все данные хранились в удобном видео. С помощью данной базы в режиме реального времени можно было бы отслеживать статистику согласно полномочиям пользователя. В планах было унифицировать данный проект для большего количества устройств, даже для мобильных телефонов, так как во многих случаях смартфон является самым простым и доступным вариантом для создания пункта наблюдения за точкой. В более далеком будущем с развитием технологий и ростом уровня мощностей, такую систему возможно было бы наделить полномочиями разделять людей по признакам, например по гендеру или возрасту. Возможность запоминать, какие люди проходили по этой точке, и их дальнейшей идентификацией, смогло бы повысить точность статистики.

Искусственный интеллект является очень мощным и гибким инструментом, который возможно реализовать для решения практически любых задач. Но самым серьезным вызовом для всего сообщества является

решение проблемы предоставления аргументации для того или иного результата работы нейронной сети.

Список литературы.

1. Comparison of YOLO v3, Faster R-CNN, and SSD for Real-Time Pill Identification - https://assets.researchsquare.com/files/rs-668895/v1_covered.pdf?c=1631875157
2. Росстат - <https://realty.rbc.ru/news/63e4e3809a794727880c079a>
3. Как работает Object Tracking на YOLO и DeepSort - <https://habr.com/ru/articles/514450/>
4. Ultralytics YOLOv8 - <https://habr.com/ru/articles/710016/>
5. Сравнение производительности моделей обнаружения объектов YOLO v5, v6 и v7 - <https://habr.com/ru/articles/702790/>
6. <https://cyberleninka.ru/article/n/sravnenie-yolo-v5-i-faster-r-cnn-dlya-obnaruzheniya-lyudey-na-izobrazhenii-v-potokovom-rezhime>
7. Компьютерное зрение: технологии, компании, тренды - <https://rdc.grfc.ru/2021/04/analytics-computer-vision/>