1. Характеристика проблемы обнаружения посторонних объектов в сыпучих материалах на основе методов машинного обучения в системах автоматического контроля
   1. Сравнительный анализ методов и средств решения задачи локализации объектов в системах автоматического контроля
      1. Анализ методов машинного обучения

Существует множество методов классификации, которые используют различный математический аппарат и различные подходы при реализации. Однако эффективность этих методов зависит от конкретной решаемой задачи.

Можно выделить следующие типы методов классификации: вероятностные, метрические, логические, линейные, логическая регрессия. Обобщенно опишем некоторые из них, указывая преимущества и недостатки каждого из них.

Метод Байеса (Naive Bayes, NB) относится к вероятностным методам классификации. Преимущества метода состоит в следующем: высокая скорость работы, поддержка инкрементного обучения, простая реализация алгоритма в виде программы, легкая интерпретируемость результатов работы алгоритма. Несмотря на приведенные достоинства, метод Байеса имеет так же и минусы в своей реализации. Относительно низкое качество классификации и неспособность учитывать зависимость результата классификации от сочетания признаков являются главными недостатками этого метода.

Метод k ближайших соседей (k Nearest Neighbors, KNN) относится к метрическим методам и считается простейшим классификатором. Объект присваивается тому классу, который является наиболее распространенным среди соседей данного элемента. Достоинства данного метода: простая реализация, проработанная теоретическая база, адаптация под нужную задачу выбором метрики или ядра, интерпретируемость. К недостаткам относятся: недостаточная производительность в реальных задачах, так как число соседей, используемых для классификации, будет достаточно большим; трудность в наборе подходящих весов и определением, какие признаки необходимы для классификации; зависимость от выбранной метрики расстояния между примерами.

Метод опорных векторов (Support Vector Machine, SVM) является линейным методом классификации, в настоящее время призван одним из лучших. Потенциальные недостатки метода опорных векторов заключается в следующем: невозможность калибровки вероятности попадания в определенный класс, подходит только для решения задач с 2 классами, параметры модели сложно интерпретировать.

Метод деревьев решений (Decision Trees, DT) относится к логическим методам классификации. Деревом решений называют ациклический граф, по которому производится классификация документов, описанных набором признаков. Каждый узел дерева содержит условие ветвления по одному из признаков. У каждого узла столько ветвлений, сколько значений имеет выбранный признак. Главным преимуществом метода является высокая производительность обучения и прогнозирования, такие деревья решений можно легко визуализировать и интерпретировать.

Еще одним из методов, позволяющий решить задачу классификации это нейронные сети. Нейронные сети активно используются в связи с появлением больших объемов данных и больших вычислительных возможностей. Сети с прямой связью являются универсальным средством аппроксимации функций, что позволяет их использовать при решении задач классификации. Их эффективность достаточно высока, потому что они генерируют фактически большое число регрессионных моделей (которые используются в решении задач классификации статистическими методами). Однако, любой метод, основанный на нейронных сетях, никогда не даст классификатор нужного качества, если имеющий набор примеров не будет достаточно полным для той задачи, с которой придется работать в системе.

Компьютерная нейронная сеть, даже самая простая по своей структуре, очень схожа с биологической нейронной сетью (ее еще называю нервной сетью). Данная сеть состоит из огромного числа нейронов (приблизительно 10 000 000 000), связанных друг с другом. Один такой нейрон представлен на рисунке \*.

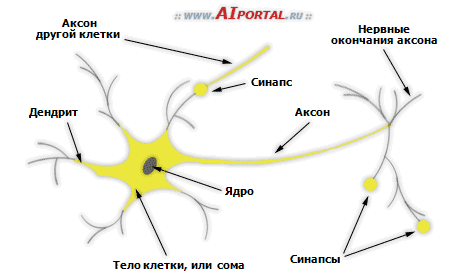


Рис.\*. Нейрон в биологической нейронной сети

Нейроны - это специальная клетки, способные распространять электрохимические сигналы. Нейрон имеет разветвленную структуру ввода информации (дендриты), ядро и разветвляющийся выход (аксон). Аксоны клетки соединяются с дендритами других клеток с помощью синапсов. При активации нейрон посылает электрохимический сигнал по своему аксону. Через синапсы этот сигнал достигает других нейронов, которые могут в свою очередь активироваться. Нейрон активируется тогда, когда суммарный уровень сигналов, пришедших в его ядро из дендритов, превысит определенный уровень (порог активации).

Благодаря таким нейронам человеческий мозг способен мыслить, рассуждать, решать задачи.

В свою очередь искусственный нейрон опишем следующим образом:

* нейрон получает входные сигналы (исходные данные либо выходные сигналы других нейронов нейронной сети) через несколько входных каналов (их число зависит от сложности и типа нейронной сети). Каждый входной сигнал проходит через соединение, имеющее определенную интенсивность (ее называют «весом»). С каждым нейроном связано определенное пороговое значение. Вычисляется взвешенная сумма входов, из нее вычитается пороговое значение и в результате получается величина активации нейрона (она также называется пост-синоптическим потенциалом нейрона - PSP).
* сигнал активации преобразуется с помощью функции активации (или передаточной функции) и в результате получается выходной сигнал нейрона.

**Функция активации** определяет выходное значение нейрона в зависимости от результата взвешенной суммы входов и порогового значения.

В разных сетях используют разные виды таких функций:

* ступенчатая функция, это такая функция, которая на выходе явно говорит нам: активный нейрон, или нет. Примером такой функции может быть Функция Хевисайда (единичная ступенчатая функция, функция единичного скачка, включённая единица, «ступенька»):

|  |  |
| --- | --- |
|  | (1) |

График такой функции представлен ан рисунке \*.

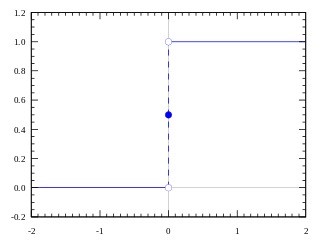


Рис.\*. Ступенчатая функция

Такие функции используют для бинарных классификаторов – моделях, в которых нам достаточно получить ответ «да» или «нет».

* Линейная функция – функция, которая представляет собой прямую линию и пропорциональна входу (то есть взвешенной сумме на этом нейроне). График функции представлен на рис.\*. Такой выбор активационной функции позволяет получать спектр значений, а не только бинарный ответ. Можно соединить несколько нейронов вместе и, если более одного нейрона активировано, решение принимается на основе применения операции max. Пример линейной функции:

|  |  |
| --- | --- |
|  | (2) |

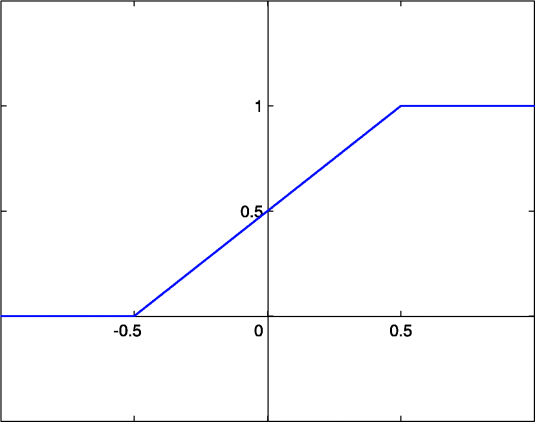


Рис.\*.График линейной функции

* Сигмоида - это гладкая монотонная возрастающая нелинейная функция, имеющая форму буквы «S», которая часто применяется для «сглаживания» значений некоторой величины. Сигмоида действительно выглядит подходящей функцией для задач классификации. Она стремиться привести значения к одной из сторон кривой (например, к верхнему при х=2 и нижнему при х=-2). Такое поведение позволяет находить четкие границы при предсказании. Функция сигмоида выглядит следующим образом:

|  |  |
| --- | --- |
|  | (3) |
| сигмоида - функция активации  Рис.\*. График сигмоида |  |

* Гиперболический тангенс – функция, схожая с сигмоидом, она так же не линейна и отлично подходит для комбинации слоев нейронной сети (график функции представлен на рис.\*). Градиент тангенциальной функции больше, чем у сигмоиды (производная круче). Выбор данной функции как активационной полностью зависит от требуемой амплитуды градиента. Пример данной функции:

|  |  |
| --- | --- |
|  | (4) |

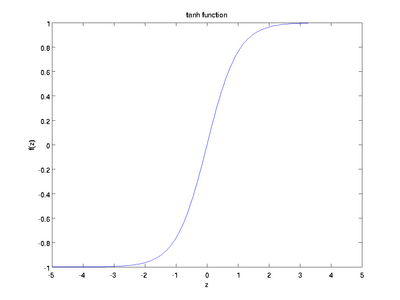


Рис.\*. График гиперболического тангенса

* ReLu – нелинейная функция, особенностью которой является ее область допустимых значений – , поэтому такая активационная функция может «взорваться». Вид функции следующий:

|  |  |
| --- | --- |
|  | (5) |

Также, часть ReLu представляет из себя горизонтальную линию (для отрицательных значений X), градиент на этой части равен 0. Из-за равенства нулю градиента, веса не будут корректироваться во время спуска. Это означает, что пребывающие в таком состоянии нейроны не будут реагировать на изменения в ошибке/входных данных (просто потому, что градиент равен нулю, ничего не будет меняться). Такое явление называется проблемой умирающего ReLu (Dying ReLu problem). Из-за этой проблемы некоторые нейроны просто выключатся и не будут отвечать, делая значительную часть нейросети пассивной. ReLu менее требовательно к вычислительным ресурсам, чем гиперболический тангенс или сигмоида, так как производит более простые математические операции. Поэтому имеет смысл использовать ReLu при создании глубоких нейронных сетей.

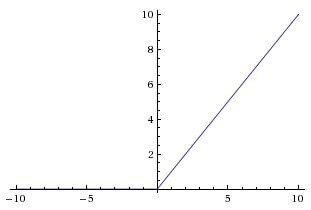


Рис.\*. График функции Relu

Простейшая сеть имеет структуру прямой передачи сигнала: Сигналы проходят от входов через скрытые элементы и в конце концов приходят на выходные элементы. Такая структура имеет устойчивое поведение. Если же сеть рекуррентная (т.е. содержит связи, ведущие назад от более дальних к более ближним нейронам), то она может быть неустойчива и иметь очень сложную динамику поведения. Рекуррентные сети представляют большой интерес для исследователей в области нейронных сетей, однако при решении практических задач, по крайней мере до сих пор, наиболее полезными оказались структуры прямой передачи.

Типичный пример сети с прямой передачей сигнала показан на рис.\*.

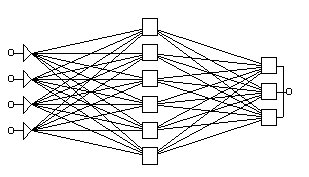


Рис.\*. Схема сети с прямой передачей сигнала

Нейроны регулярным образом организованы в слои. Входной слой служит просто для ввода значений входных переменных. Каждый из скрытых и выходных нейронов соединен со всеми элементами предыдущего слоя. Можно было бы рассматривать сети, в которых нейроны связаны только с некоторыми из нейронов предыдущего слоя; однако, для большинства приложений сети с полной системой связей предпочтительнее.

Начиная с 2016 года нейронные сети шагнули вперед в своем развитии по следующим направлениям [4]:

* системы распознавания и классификации объектов на изображениях;
* голосовые интерфейсы взаимодействия для интернета вещей;
* системы мониторинга качества обслуживания в колл-центрах;
* системы выявления неполадок (в том числе, предсказывающие время технического обслуживания), аномалий, кибер-физических угроз;
* системы интеллектуальной безопасности и мониторинга;
* замена ботами части функций операторов колл-центров;
* системы видеоаналитики;
* самообучающиеся системы, оптимизирующие управление материальными потоками или расположение объектов (на складах, транспорте);
* интеллектуальные, самообучающиеся системы управления производственными процессами и устройствами (в том числе, робототехнические);
* появление систем универсального перевода «на лету» для конференций и персонального использования;
* появление ботов-консультантов технической поддержки или персональных ассистентов, по функциям близким к человеку.

Именно нейронные сети позволят решить задачу распознавание и классификация объектов на изображении, для дальнейшего подсчёта вагонов, входящих в состав. Анализ типов нейронных сетей приведен в следующем пункте.

* + 1. Анализ типов нейронных сетей

Нейронные сети сейчас постоянно получаю архитектурное развитие, создаются новые, совершенствуются старые. В основном выделяют 13 типов сетей, рассмотренные далее [1].

В каждой модели нейронной сети нейроны обозначены в соответствии с рис.3.



Рис.3. Обозначение нейронов в моделях нейронных сетей



Рис.4. Нейронные сети прямого распространения

Нейронные сети прямого распространения (feed forward neural networks, FF или FFNN) и перцептроны (perceptrons, P) очень прямолинейны, они передают информацию от входа к выходу (рис.4, 5). Нейронные сети часто описываются в виде слоёного торта, где каждый слой состоит из входных, скрытых или выходных клеток. Клетки одного слоя не связаны между собой, а соседние слои обычно полностью связаны. Самая простая нейронная сеть имеет две входных клетки и одну выходную, и может использоваться в качестве модели логических вентилей. FFNN обычно обучается по методу обратного распространения ошибки, в котором сеть получает множества входных и выходных данных. Этот процесс называется обучением с учителем, и он отличается от обучения без учителя тем, что во втором случае множество выходных данных сеть составляет самостоятельно. Вышеупомянутая ошибка является разницей между вводом и выводом. Если у сети есть достаточное количество скрытых нейронов, она теоретически способна смоделировать взаимодействие между входным и выходными данными. Практически такие сети используются редко, но их часто комбинируют с другими типами для получения новых.

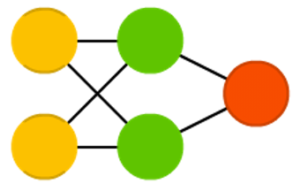


Рис.5. Нейронные сети радиально-базисных функций

Сети радиально-базисных функций (radial basis function, RBF) — это FFNN, которая использует радиальные базисные функции как функции активации (рис.3.). В остальном – это нейронная сеть прямого распространения.

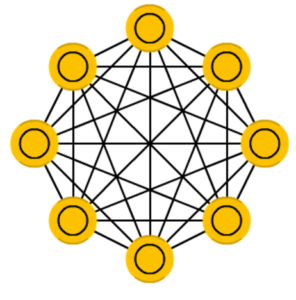


Рис.6. Нейронная сеть Хопфилда

Нейронная сеть Хопфилда (Hopfield network, HN) — это полносвязная нейронная сеть с симметричной матрицей связей (рис.6). Во время получения входных данных каждый узел является входом, в процессе обучения он становится скрытым, а затем становится выходом. Сеть обучается так: значения нейронов устанавливаются в соответствии с желаемым шаблоном, после чего вычисляются веса, которые в дальнейшем не меняются. После того, как сеть обучилась на одном или нескольких шаблонах, она всегда будет сводиться к одному из них (но не всегда — к желаемому). Она стабилизируется в зависимости от общей «энергии» и «температуры» сети. У каждого нейрона есть свой порог активации, зависящий от температуры, при прохождении которого нейрон принимает одно из двух значений (обычно -1 или 1, иногда 0 или 1). Такая сеть часто называется сетью с ассоциативной памятью; как человек, видя половину таблицы, может представить вторую половину таблицы, так и эта сеть, получая таблицу, наполовину зашумленную, восстанавливает её до полной.

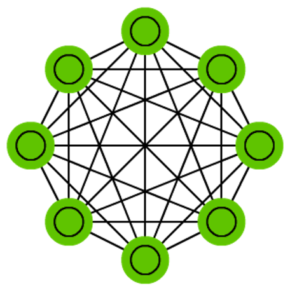


Рис.7. Цепи Маркова

Цепи Маркова (Markov chains, MC или discrete time Markov Chains, DTMC) — это предшественники машин Больцмана (BM) и сетей Хопфилда (HN) (рис.7). Их смысл можно объяснить так: каковы мои шансы попасть в один из следующих узлов, если я нахожусь в данном? Каждое следующее состояние зависит только от предыдущего. Хотя на самом деле цепи Маркова не являются НС, они весьма похожи. Также цепи Маркова не обязательно полносвязны.

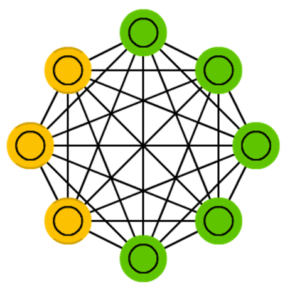


Рис.8. Машина Больцмана

Машина Больцмана (Boltzmann machine, BM) очень похожа на сеть Хопфилда, но в ней некоторые нейроны помечены как входные, а некоторые — как скрытые (рис.8). Входные нейроны в дальнейшем становятся выходными. Машина Больцмана — это стохастическая сеть. Обучение проходит по методу обратного распространения ошибки или по алгоритму сравнительной расходимости. В целом процесс обучения очень похож на таковой у сети Хопфилда.

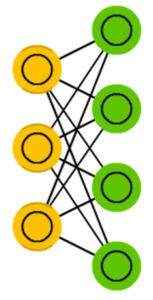


Рис.9. Ограниченная машина Больцмана

Ограниченная машина Больцмана (restricted Boltzmann machine, RBM) удивительно похожа на машину Больцмана и, следовательно, на сеть Хопфилда (рис.9). Единственной разницей является её ограниченность. В ней нейроны одного типа не связаны между собой. Ограниченную машину Больцмана можно обучать как FFNN, но с одним нюансом: вместо прямой передачи данных и обратного распространения ошибки нужно передавать данные сперва в прямом направлении, затем в обратном. После этого проходит обучение по методу прямого и обратного распространения ошибки.

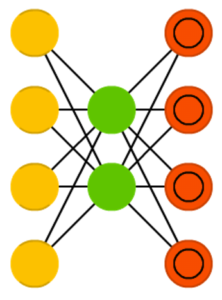


Рис.10. Автокодировщик

Автокодировщик (autoencoder, AE) чем-то похож на FFNN, так как это скорее другой способ использования FFNN, нежели фундаментально другая архитектура (рис.10). Основной идеей является автоматическое кодирование (в смысле сжатия, не шифрования) информации. Сама сеть по форме напоминает песочные часы, в ней скрытые слои меньше входного и выходного, причём она симметрична. Сеть можно обучить методом обратного распространения ошибки, подавая входные данные и задавая ошибку равной разнице между входом и выходом.

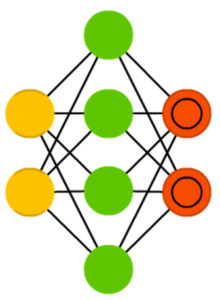


Рис.11. Разреженный автокодировщик

Разреженный автокодировщик (sparse autoencoder, SAE) — в каком-то смысле противоположность обычного (рис.11). Вместо того, чтобы обучать сеть отображать информацию в меньшем «объёме» узлов, мы увеличиваем их количество. Вместо того, чтобы сужаться к центру, сеть там раздувается. Сети такого типа полезны для работы с большим количеством мелких свойств набора данных. Если обучать сеть как обычный автокодировщик, ничего полезного не выйдет. Поэтому кроме входных данных подаётся ещё и специальный фильтр разреженности, который пропускает только определённые ошибки.

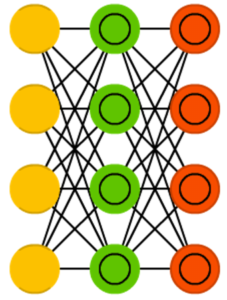


Рис.12. Вариационные автокодировщики

Вариационные автокодировщики (variational autoencoder, VAE) обладают схожей с AE архитектурой, но обучают их иному: приближению вероятностного распределения входных образцов (рис.12). В этом они берут начало от машин Больцмана. Тем не менее, они опираются на байесовскую математику, когда речь идёт о вероятностных выводах и независимости, которые интуитивно понятны, но сложны в реализации. Если обобщить, то можно сказать что эта сеть принимает в расчёт влияния нейронов. Если что-то одно происходит в одном месте, а что-то другое — в другом, то эти события не обязательно связаны, и это должно учитываться.

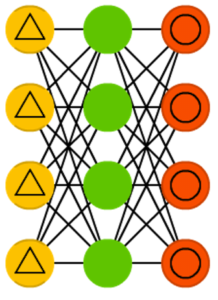


Рис.13. Шумоподавляющие автокодировщики

Шумоподавляющие автокодировщики (denoising autoencoder, DAE) — это AE, в которые входные данные подаются в зашумленном состоянии (рис.13). Ошибку мы вычисляем так же, и выходные данные сравниваются с зашумленными. Благодаря этому сеть учится обращать внимание на более широкие свойства, поскольку маленькие могут изменяться вместе с шумом.

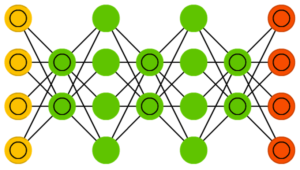


Рис.14. Сеть типа «deep belief»

Сеть типа «deep belief» (deep belief networks, DBN) — это название, которое получил тип архитектуры, в которой сеть состоит из нескольких соединённых RBM или VAE (рис.14). Такие сети обучаются поблочно, причём каждому блоку требуется лишь уметь закодировать предыдущий. Такая техника называется «жадным обучением», которая заключается в выборе локальных оптимальных решений, не гарантирующих оптимальный конечный результат. Также сеть можно обучить (методом обратного распространения ошибки) отображать данные в виде вероятностной модели. Если использовать обучение без учителя, стабилизированную модель можно использовать для генерации новых данных.

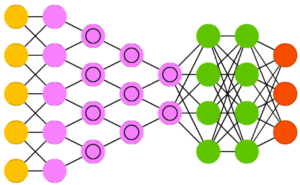


Рис.15. Свёрточные нейронные сети

Свёрточные нейронные сети (convolutional neural networks, CNN) и глубинные свёрточные нейронные сети (deep convolutional neural networks, DCNN) сильно отличаются от других видов сетей (рис.15). Обычно они используются для обработки изображений, реже для аудио. Типичным способом применения CNN является классификация изображений: если на изображении есть кошка, сеть выдаст «кошка», если есть собака — «собака». Такие сети обычно используют «сканер», не парсящий все данные за один раз. Например, если у вас есть изображение 200×200, вы не будете сразу обрабатывать все 40 тысяч пикселей. Вместо это сеть считает квадрат размера 20 x 20 (обычно из левого верхнего угла), затем сдвинется на 1 пиксель и считает новый квадрат, и т.д. Эти входные данные затем передаются через свёрточные слои, в которых не все узлы соединены между собой. Эти слои имеют свойство сжиматься с глубиной, причём часто используются степени двойки: 32, 16, 8, 4, 2, 1. На практике к концу CNN прикрепляют FFNN для дальнейшей обработки данных. Такие сети называются глубинными (DCNN).

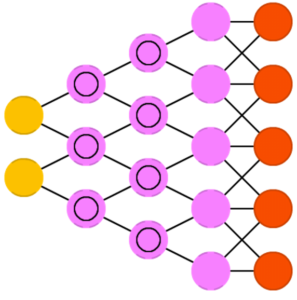


Рис.16. Развёртывающие нейронные сети

Развёртывающие нейронные сети (deconvolutional networks, DN), также называемые обратными графическими сетями, являются обратным к свёрточным нейронным сетям (рис.16). Представьте, что вы передаёте сети слово «кошка», а она генерирует картинки с кошками, похожие на реальные изображения котов. DNN тоже можно объединять с FFNN. Стоит заметить, что в большинстве случаев сети передаётся не строка, а какой бинарный вектор: например, <0, 1> — это кошка, <1, 0> — собака, а <1, 1> — и кошка, и собака.

Из всего многообразия видов нейронных сетей, лучшим для решения поставленной задачи являются сверточные нейронные сети, которые рассмотрены в следующем пункте.

* + 1. Анализ архитектур многослойных сверточных нейронных сетей

LeNet – первая сверточная нейронная сеть, предложенная Яном Лекуном, состоит из трёх типов слоёв: свёрточные (convolutional) слои, субдискретизирующие (subsampling, подвыборка) слои и слои "обычной" нейронной сети – перцептрона. Первые два типа слоёв (свёрточные, субдискретизирующие), чередуясь между собой, формируют входной вектор признаков для многослойного перцептрона. Сеть можно обучать с помощью градиентных методов. Данная архитектура без значительных изменений до сих пор в большинстве случаев используется при решении прикладных задач медицинской диагностики на изображениях и видеоданных.

AlexNet (Алекс Крижевский) – данная сеть очень похожа на сеть LeNet, однако отличалась от неё более массивной и сложной архитектурой. Также она имела всего лишь один сверточный слой и несколько слоев субдискретизации по принципу выбора максимального значения. Сверточная нейронная сеть AlexNet предназначена для распознавания объектов любой сложности на больших изображениях размером 224 х 224.

ZF Net (Метью Цейлер, Робер Фергюс) - Данная сеть представляет собой модификацию сети AlexNet, основная особенность которой заключается в более удачном наборе параметров сети: увеличение размеров внутренних сверточных слоев сети, а также уменьшение размеров смещения и размеров фильтров в первом сверточном слое.

GoogLeNet (Google) - данная сверточная сеть является очень глубокой – до 22 слоев. Но, несмотря на это, имеет в 10 раз меньше параметров, чем сеть AlexNet, что положительно сказывается на производительности и расходовании памяти. Также в ней используются малые размеры фильтров, а слой субдискретизации реализован по принципу выбора среднего значения.

VGGNet (Карен Симонян, Эндрю Зиссерман) - Данная СНС, также, как и GoogLeNet, является очень глубокой (до 16 слоев) и состоит из большого количества чередующихся слоев свертки и субдискретизации, имеющих малые размеры (3 х 3 – размер фильтров сверточного слоя, 2 х 2 – размер окон слоя субдискретизации). Отрицательной стороной данной сети является то, что она хранит до 140 миллионов параметров, что делает её громоздкой и низко-производительной.

ResNet (Microsoft Research Asia) - архитектура данной сети предполагает большое количество сверточных слоев, содержащих большое количество (до 512) фильтров малого размера (3 х 3). Глубина сети может достигать 152 слоев. На примере данной сети было установлено, что СНС может использовать только сверточные слои и качество распознавания значительно увеличивается при увеличении глубины сети. Данная сеть является одной из наиболее эффективных сверточных нейронных сетей на сегодняшний день.

Mask-RCNN (UC Berkley) – типичная свёрточная нейронная сеть, позволяющая не только распознавать и классифицировать объект, но и выделять его контур («маску»). Данная сеть является улучшенной версией R-CNN сети, в качестве которой использовалась так же готовая архитектура — CaffeNet (AlexNet). Такие нейросети, как и другие для набора изображений ImageNet, проводят классификацию на 1000 классов. R-CNN разрабатывалась для детектирования объектов меньшего количества классов (N= 20 или 200), поэтому последний классификационный слой CaffeNet был заменён на слой с N+1 выходами (с дополнительным классом для фона).

Несмотря на то, что CNN тренировалась на распознавание N+1 классов, в итоге она использовалась только для извлечения, фиксированного 4096-размерного вектора признаков. Непосредственным определением объекта на изображении занимались N линейных SVM, каждый из которых проводил бинарную классификацию по своему типу объектов, определяя есть ли такой в переданном регионе или нет. В оригинальном документе вся процедура иллюстрируется такой схемой, представленной на рис.\*.

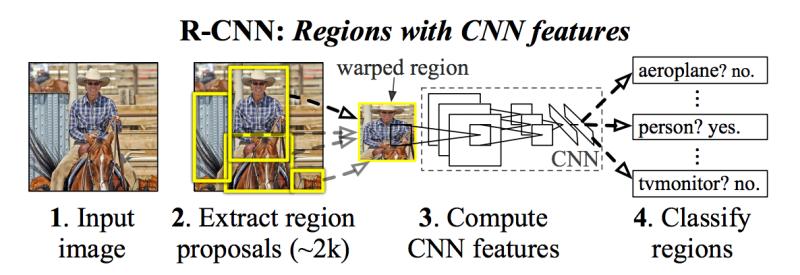


Рис.\*. Схема работы R-CNN сети

Таким образом, процедуру детектирования объектов сетью R-CNN можно разделить на следующие шаги:

* выделение регионов-кандидатов при помощи Selective Search:
* преобразование региона в размер, принимаемый CNN CaffeNet;
* получение при помощи CNN 4096-размерного вектора признаков;
* проведение N бинарных классификаций каждого вектора признаков при помощи N линейных SVM;
* линейная регрессия параметров рамки региона для более точного охвата объекта.

Mask R-CNN развивает архитектуру R-CNN путём добавления ещё одной ветки, которая предсказывает положение маски, покрывающей найденный объект, и, таким образом решает уже задачу instance segmentation (сегментация экземпляра). Маска представляет собой просто прямоугольную матрицу, в которой 1 на некоторой позиции означает принадлежность соответствующего пикселя объекту заданного класса, 0 — что пиксель объекту не принадлежит.

Для решения задачи распознавания и классификации была выбрана архитектура Mask R-CNN, которая позволит быстро производить анализ видеопотока на не мощном оборудовании, например, видеокарте NVIDIA GTX 960, примерно со скоростью 20-25 кадров в секунду, с минимальной вероятностью появления ложных распознаваний. Дополнительной особенностью станет более точное выделение посторонних объектов в кадре.

Следующим пунктом рассмотрены нынешние аппаратно-программные комплексы (АПК) на рынке, работающие на основе нейронных сетей, со схожей архитектурой.

* + 1. Анализ существующих аппаратно-программных средств решения

Нейросетевые алгоритмы находят широкое применение в вычислительной практике. В частности, особое внимание уделяется классификации и кластеризации изображений, распознаванию речи и изображений, прогнозу финансовых показателей, искусственному синтезу речи, аппроксимации функционалов, совершенствованию методов электроимпедансной и магнитоиндукционной томографии, анализу данных (data mining) [9].

Известным ограничением развития нейросетевых алгоритмов следует признать высокие вычислительные затраты на реализацию таких методов [2]. К традиционным способам решения данной проблемы относят организацию параллельных и распределенных вычислений на специализированном аппаратном обеспечении, таком как нейронные чипы, систолические нейропроцессоры, ПЛИС, распределенные кластерные системы [3], GRID-технологии. Появившийся недавно аппаратно-программный комплекс CUDA (Compute Unified Device Architecture) позволяет использовать процессоры видеокарт (GPU) как ускорители научных и инженерных расчетов и проводить вычисления, по эффективности сравнимые с современными кластерными системами [4][5]. Принципиальное отличие архитектур состоит в следующем: исполнение команд на кластере происходит в стиле MIMD (много потоков команд, много потоков данных – когда набор процессоров независимо выполняет различные наборы команд, обрабатывающих различные наборы данных), а в среде CUDA характеризуется стилем SIMD (один поток команд, много потоков данных – когда несколько процессоров исполняют одну и ту же команду над разными данными, обычно элементами массива). Особенностью оборудования, поддерживающего технологию CUDA, является возможность обеспечивать на порядок большую (по сравнению с кластерами) пропускную способность при работе с памятью. В частности, при вычислении кулоновской ионизации сверхбольших структур, таких как вирусы, может потребоваться несколько дней работы кластера средних размеров, в то время как решение той же задачи на персональном компьютере, оснащенном видеокартой с поддержкой CUDA, можно провести менее чем за час [6].

В настоящее время аппаратная часть CUDA присутствует во многих (в том числе и бюджетных) видеокартах NVIDIA. Таким образом, для полноценной работы с нейронными сетями следует установить ПО NVIDIA CUDA, которая обеспечивает возможность выполнения на GPU любых вычислений, либо библиотеку cuDNN, которая специально разработана для обучения глубоких нейронных сетей. Она содержит оптимизированные для GPU реализации свёрточных и рекуррентных сетей, различных функций активации (полулинейная, сигмоидальная, гиперболический тангенс, softmax), алгоритма обратного распространения ошибки и т.п. cuDNN позволяет обучать нейронные сети на GPU в несколько раз быстрее, чем просто CUDA. Библиотека cuDNN содержит оптимизированные для GPU функции обучения нейронных сетей. Она позволяет обучать нейронную сеть в несколько раз быстрее, чем с использованием CUDA, и в несколько десятков раз быстрее, чем на CPU [4][6].

OpenCV — библиотека алгоритмов компьютерного зрения, обработки изображений и численных алгоритмов общего назначения с открытым кодом. Реализована на C/C++, также разрабатывается для Python, Java, Ruby, Matlab, Lua и других языков. Может свободно использоваться в академических и коммерческих целях — распространяется в условиях лицензии BSD [5].

Keras — открытая нейросетевая библиотека, написанная на языке Python. Она представляет собой надстройку над фреймворками Deeplearning, TensorFlow и Theano. Нацелена на оперативную работу с сетями глубинного обучения, при этом спроектирована так, чтобы быть компактной, модульной и расширяемой. Она была создана как часть исследовательских усилий проекта ONEIROS (англ. Open-ended Neuro-Electronic Intelligent Robot Operating System), а ее основным автором и поддерживающим является Франсуа Шолле, инженер Google. Эта библиотека содержит многочисленные реализации широко применяемых строительных блоков нейронных сетей, таких как слои, целевые и передаточные функции, оптимизаторы, и множество инструментов для упрощения работы с изображениями и текстом [5][6].

TensorFlow — открытая программная библиотека для машинного обучения, разработанная компанией Google для решения задач построения и тренировки нейронной сети с целью автоматического нахождения и классификации образов, достигая качества человеческого восприятия [6].

Автомаршал – программное обеспечение для автоматического распознавания номеров автомобилей в транспортном потоке и в пунктах контроля (рис.17) [18]. Применяется для автоматизации работы парковок, КПП и проходных предприятий, логистических и складских комплексов, учета транспорта на автомагистралях, весовых, автомойках и заправочных станциях, а также для управления доступом автомобилей на территорию многоквартирных жилых домов и загородных поселков.

Traffix - программное обеспечение предназначено для подсчета интенсивности и определения состава трафика по видеоизображению (рис.18) [18]. Видео поступает непосредственно с обзорных и специальных камер онлайн, либо может быть загружено из видеофайла. В результате анализа программа выдает статистику по количеству транспортных средств и их категориям.

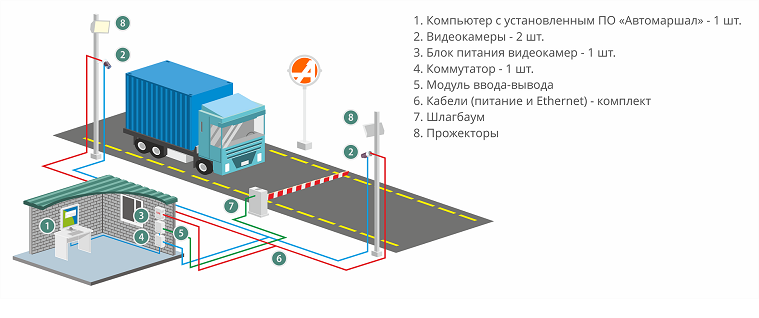


Рис.17. Модель АПК «Автомаршал»

Адаптивная смарт-камера VIRIS с высоким уровнем распознавания номеров автомобилей [18]. Предлагаемая технологическая новинка объединяет в себе возможности видеокамеры и ПК, а специальный интеллектуальный модуль обеспечивает адаптацию устройства к различным изменениям внешней среды.

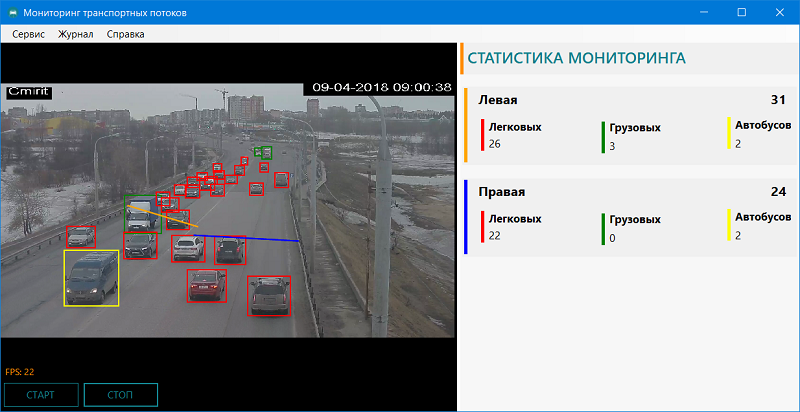


Рис.18. Программа Traffix

АРСИС (ARSCIS) – это семейство решений по идентификации вагонов, управлению отгрузкой продукции ж/д транспортом, коммерческому осмотру вагонов, контролю передвижения и местонахождения вагонов, построенное на платформе системы распознавания номеров вагонов и решающее задачи служб метрологии, логистики и безопасности (рис.19) [18].

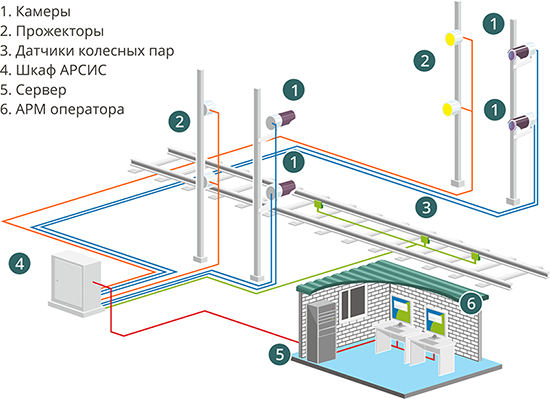


Рис.19. АПК «АРСИС»

Road Traffic Analysis – программное обеспечение, предназначенное для анализа дорожного трафика, по видеоизображени: интенсивность движения, скорость автомобилей, плотность трафика и т.п. (рис.20) [18].

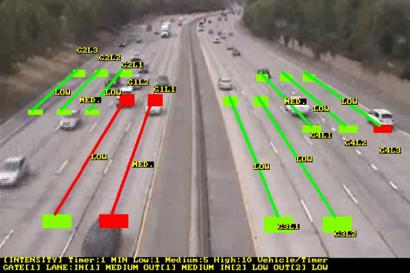


Рис.20. Программа Road Traffic Analysis

Все эти и другие программные решения выполняют конкретную задачу, например, распознавание номеров на цистернах, или классификация номеров. Нет на рынке готово решения, которое бы позволило вести подсчет вагонов по видеоизображения.

Для разработки программного обеспечения, решающего поставленную задачу, необходимо проанализировать объекты, которые будут находиться в кадре, получаемое с обзорных видеокамер. Описание представлено в следующем пункте.

* 1. Описание объектов на изображении для решения задачи

Шихтовый двор при агломерационном производстве в ЧерМК занимает примерно 400м2 открытого пространства, на которое сбрасывают сыпучий материал самосвалы.

В компании используются самосвалы марки AMKAR-6589E, размеры которого составляют 8260х255х3220 мм, а база - 3600+1440.

Автомобиль-самосвал имеет стальную платформу прямоугольного сечения. Основание платформы с несущим каркасом и креплением гидроцилиндра в передней части. Запоры заднего борта автоматические, открываются при подъеме платформы. Опрокидывающее устройство платформы гидравлическое с приводом от насоса через коробку отбора мощности. Управление подъемом электропневматическое из кабины. Самосвальная установка оснащена тентом и механизмом сбора тента. Платформа имеет обогрев днища по 4-х канальной системе, дополнительный обогрев в передней части кузова и задних стоек.

Металлическая шихта представляет собой твердый, похожий на камень, материал, размерами от 50 мм до 200 мм и, как правило, серого цвета (рис.21), основу которого составляет чугун – твердый или жидкий, а также стальной лом.



Рис.21. Шихта в близи

На дворе шихта формируется кучами благодаря бульдозерам на базе гусеничных тракторов под управлением машинистов – сотрудников агломерационного производства. Такая машина имеет следующие технические характеристики:

* номинальную мощность, достигающую 603 кВт;
* эксплуатационную массу до 155 т;
* длину отвала до 5500 мм, высоту с козырьком до 2300 мм;
* высоту подъема рабочего органа (по нижней кромке) до 1780 мм, заглубление до 800 мм;
* массу с навесным оборудованием до 32 т.

По мимо шихты, как было описано в проблеме, вместе с полезным сыпучим материалом, самосвалы привозят и посторонние предметы разного происхождения и размеров (рис.22). Например, это может быть тонкие обрезки листов металла, гнутые профиля, пластиковые изделия, например, бутылки, обрывки оградительных лент и т.д.



Рис.22. Посторонние предметы в шихте

* 1. Постановка задачи, разработка требований к информационному и программному обеспечению
     1. Постановка задачи

Ключ.фраза – Основным назначением системы …

В перспективе…

Для реализации в данной работе сформулируем задачу для начального этапа…

* + 1. Разработка требований к информационному и аппаратно-программному обеспечению

Требования к функциональным характеристикам:

* выбор языка и среды программирования в соответствии с ИТконцепцией организации-заказчика, по возможности среди свободного программного обеспечения;
* а

Входные данные:

* аа
* а

Выходные данные:

* 1. Выводы по разделу 1