|  |  |
| --- | --- |
| Институт (факультет) | Информационных технологий |
| Кафедра | Математическое и программное обеспечение ЭВМ |

КУРСОВОЙ ПРОЕКТ

|  |
| --- |
| по модулю Проектная деятельность |

|  |  |
| --- | --- |
| на тему | Разработка модуля сопровождения сцепок между вагонами посредством нейронных |
| сетей | |

|  |
| --- |
| Выполнил студент группы 1АПм-02-11оп |
| *группа* |
| направления подготовки (специальности) |
| 15.04.04 Автоматизация технологических процессов и производств, программа Аддитивные цифровые системы и технологии |
| *шифр, наименование* |
| **Белов Антон Сергеевич** |
| *фамилия, имя, отчество* |

|  |
| --- |
| Руководитель |
| **Ершов Евгений Валентинович** |
| *фамилия, имя, отчество* |
| **доктор технических наук** |
| *должность* |

|  |
| --- |
| Дата представления работы |
| «\_\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20 \_\_\_ г. |
|  |
| Заключение о допуске к защите |
|  |
|  |
|  |
|  |
| Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_, \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| количество баллов |
| Подпись преподавателя\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

Череповец, 2019

*Год*

Оглавление

[1. Характеристика проблемы локализации объектов на основе методов машинного обучения в системах автоматического контроля 3](#_Toc536403022)

[1.1. Сравнительный анализ методов и средств решения задачи локализации объектов в системах автоматического контроля 4](#_Toc536403023)

[1.1.1. Анализ методов машинного обучения 4](#_Toc536403024)

[1.1.2. Анализ типов нейронных сетей 7](#_Toc536403025)

[1.1.3. Анализ архитектур многослойных сверточных нейронных сетей 15](#_Toc536403026)

[1.1.4. Анализ аппаратно-программных средств решения задач локализации объектов на основе свертыочных нейронных сетей 16](#_Toc536403027)

[1.2. Описание объектов на изображении для решения задачи локализации в системах автоматизированного контроля 17](#_Toc536403028)

[1.3. Постановка задачи на локализацию и формирование требований к математическому, алгоритмическому информационному и программному обеспечению 18](#_Toc536403029)

1. Характеристика проблемы локализации объектов на основе методов машинного обучения в системах автоматического контроля

С появлением персональных компьютеров, работа человека упростилась в десятки раз, благодаря автоматизации его рутинных задач. Решение математических задач, задач оптимизации, бухгалтерский учёт и многое другое, все это решает за человека программное обеспечение, установленное на его компьютере.

Нынешние технологии позволяют обрабатывать 100 Тбайт информации за считанные минуты. Это позволяет практически мгновенно анализировать нынешнею ситуацию и принимать соответствующие решения. Но данную информацию необходимо получить из окружающей среды и передать в информационную систему для дальнейшей обработки.

Например, одной из основных задач по созданию и развитию систем автоматизированного управления железнодорожными грузоперевозками является автоматизация процесса контроля передвижения объектов подвижного состава железнодорожного транспорта, включая их идентификацию. Одной из-подзадач контроля передвижения является подсчет вагонов, входящих в состав.

Нынешняя задача как правило решается с использованием датчиков магнитного типа, информационный сигнал которых формируется на основе эффекта электро­магнитной индукции, или датчика прохода колес, который работает по принципу наведении в катушке ЭДС индукции за счет изменения величины магнитного потока при проходе гребня колеса.

Недостатком первого типа датчиков является ограничение по скоростному диапазону, так как амплитуда импульса падает с уменьшением скорости состава, и при остановках колеса над датчиком напряжение практически отсутствует. К недостаткам датчиков магнитного типа можно отнести большие размеры, большое содержание меди и наличие сильного магнита. Это провоцирует вандализм и, в конечном счете, увеличивает затраты на обслуживание. Альтернативным решением является использование машинного обучения.

Машинное обучение (англ. machine learning, ML) — класс методов искусственного интеллекта, характерной чертой которых является не прямое решение задачи, а обучение в процессе применения решений множества сходных задач. Для построения таких методов используются средства математической статистики, численных методов, методов оптимизации, теории вероятностей, теории графов, различные техники работы с данными в цифровой форме. Технология машинного обучения – это учитель, который обучает компьютеры выполнению задач на основе данных, а не на базе алгоритма, заложенного программистом.

Такая технология позволит программе распознавать объекты в кадре, для будущего подсчёта количества вагонов, вне зависимости от посторонних объектов в кадре.

* 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.

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.

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

* 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 распознавание и классификация объектов на изображении позволит решить задачу подсчёта вагонов, входящих в состав.

* + 1. Анализ типов нейронных сетей

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



Рис.1. 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.1). 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.

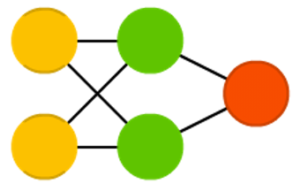


Рис.2. Нейронные сети радиально-базисных функций

Сети радиально-базисных функций (radial basis function, RBF) — это FFNN, которая использует радиальные базисные функции как функции активации (рис.2.). В остальном – это нейронная сеть прямого распространения.

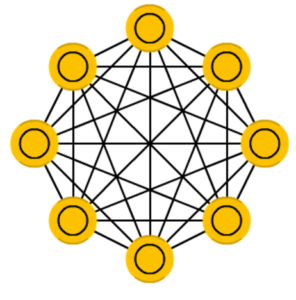


Рис.3. Нейронная сеть Хопфилда

Нейронная сеть Хопфилда (Hopfield network, HN) — это полносвязная нейронная сеть с симметричной матрицей связей. 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.

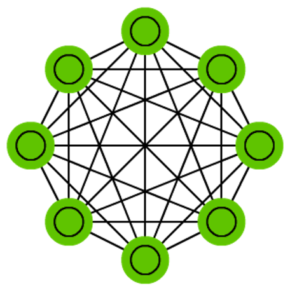


Рис.4. Цепи Маркова

Цепи Маркова (Markov chains, MC или discrete time Markov Chains, DTMC) — это предшественники машин Больцмана (BM) и сетей Хопфилда (HN). 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.

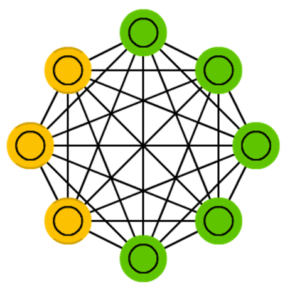


Рис.5. Машина Больцмана

Машина Больцмана (Boltzmann machine, BM) очень похожа на сеть Хопфилда, но в ней некоторые нейроны помечены как входные, а некоторые — как скрытые. 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.

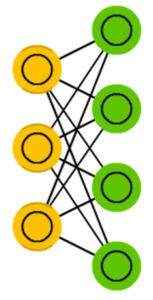


Рис.6. Ограниченная машина Больцмана

Ограниченная машина Больцмана (restricted Boltzmann machine, RBM) удивительно похожа на машину Больцмана и, следовательно, на сеть Хопфилда. 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.

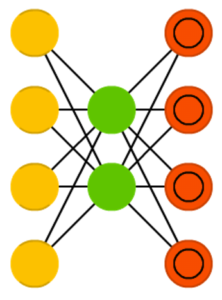


Рис.7. Автокодировщик

Автокодировщик (autoencoder, AE) чем-то похож на FFNN, так как это скорее другой способ использования FFNN, нежели фундаментально другая архитектура. 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.

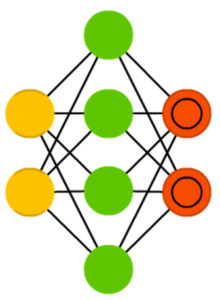


Рис.8. Разреженный автокодировщик

Разреженный автокодировщик (sparse autoencoder, SAE) — в каком-то смысле противоположность обычного. 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.

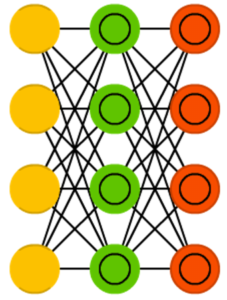


Рис.9. Вариационные автокодировщики

Вариационные автокодировщики (variational autoencoder, VAE) обладают схожей с AE архитектурой, но обучают их иному: приближению вероятностного распределения входных образцов. 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.

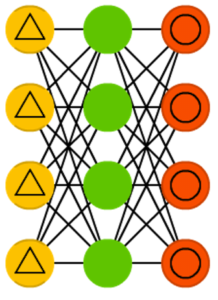


Рис.10. Шумоподавляющие автокодировщики

Шумоподавляющие автокодировщики (denoising autoencoder, DAE) — это AE, в которые входные данные подаются в зашумленном состоянии. 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.

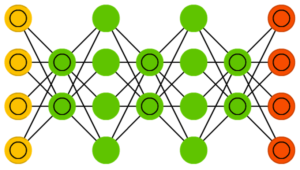


Рис.11. Сеть типа «deep belief»

Сеть типа «deep belief» (deep belief networks, DBN) — это название, которое получил тип архитектуры, в которой сеть состоит из нескольких соединённых RBM или VAE. 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.

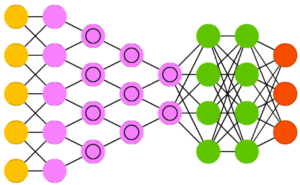


Рис.12. Свёрточные нейронные сети

Свёрточные нейронные сети (convolutional neural networks, CNN) и глубинные свёрточные нейронные сети (deep convolutional neural networks, DCNN) сильно отличаются от других видов сетей. 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).

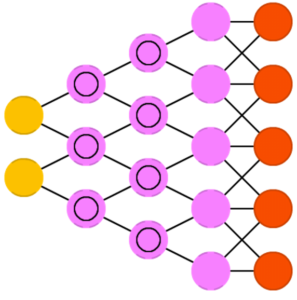


Рис.13. Развёртывающие нейронные сети

Развёртывающие нейронные сети (deconvolutional networks, DN), также называемые обратными графическими сетями, являются обратным к свёрточным нейронным сетям. 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.

* + 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.

Для решения задачи подсчета вагонов достаточно использовать архитектуру сети AlexNet, так как для решения задачи подсчёта вагонов не требуется большое число сверточных слоев.

* + 1. Анализ аппаратно-программных средств решения задач локализации объектов на основе свертыочных нейронных сетей

Автомаршал – программное обеспечение для автоматического распознавания номеров автомобилей в транспортном потоке и в пунктах контроля. 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. 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.

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

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.

Road Traffic Analysis – программное обеспечение, предназначенное для анализа дорожного трафика, по видеоизображени: интенсивность движения, скорость автомобилей, плотность трафика и т.п.

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.

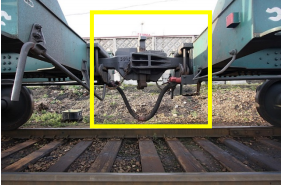
* 1. Описание объектов на изображении для решения задачи локализации в системах автоматизированного контроля

Во время проезда состава через городское депо, оператор получает картинку с нескольких смотровых камер. Одна из таких камер расположена перпендикулярно движению состава, на высоте примерно равно высоте колес вагона 950 мм, плюс высота рельс – 180 мм.

На изображение присутствует:

* фон – объекты ж/д депо, который может состоять как из кустарных растений и деревьев, так и из рядом проходящего состава;
* состав – состоит из группы вагонов, с одним или несколькими действующими локомотивами или моторными вагонами, приводящими его в движение, и имеющий установленные сигналы (звуковые и видимые), которые обозначают его голову и хвост:
  + локомотив или моторный вагон (голова) – самоходный рельсовый экипаж, предназначенный для тяги несамоходных вагонов;
  + вагон – несамоходное транспортное средство, предназначенное для движения по рельсам или иным путям и эксплуатируемое с локомотивом. Предназначено для перевозки пассажиров или иного физического груза (уголь, песок, нефть, кислота, автомобили, спецтехника и тд);
  + сцепка - неавтоматическое сцепное устройство, применяемое на железнодорожном транспорте, с целью соединения вагонов в единый состав.
  1. Постановка задачи на локализацию и формирование требований к математическому, алгоритмическому информационному и программному обеспечению

Требуется разработать аппаратно-программный комплекс для автоматизированной системы управления АРСИС, который позволит обнаруживать, сопровождать и подсчитывать на видео железнодорожную сцепку (устройство соединения вагонов между собой) (рис.\*).



Требования к аппаратно-программному комплексу:

Аппаратно-программный комплекс должен состоять из IP-камеры и программного обеспечения.

IP-камера: матрица должна быть построена по технологии CMOS, в силу своей светочувствительности, что позволит передавать видимую картинку в ночное время суток без дополнительной подсветки; разрешение матрицы не ниже 1280x720 (HD); частота кадров – 25 к/с; корпус камеры должен быть выполнен по технологии аддитивного производства SLS, основанная на послойном спекании порошковых материалов (полиамиды, пластик) с помощью луча лазера. Приблизительные размеры камеры: 70×70×170мм. Камера должна быть оборудована крепежной площадкой со резьбой 1/4 дюйма, 20 витков на дюйм, для быстрой фиксации на смотровом столбе или штативе.

Требование к программному обеспечению описаны ниже.

Анализируемые данные:

Исходные данные представляют собой видеопоток (видеофайл), полученный со смотровой IP-видеокамеры. Расположение камеры – приблизительно на уровне сцепки (1100мм). Камера располагается неподвижно под 90 градусов по отношению к ж/д пути. При таком расположении камер сцепки двигаются всегда по горизонтали слева направо или справа налево в зависимости от направления движения состава.

Требования к программному модулю:

Алгоритм должен устойчиво работать в нестационарных условиях внешней среды:

1. При любых погодных условиях.
2. В любое время суток (при условии наличия освещения).
3. Фон за вагонами может быть любым (в том числе подвижным, например, на соседнем пути могут двигаться другие вагоны, ходить люди).

Алгоритм не должен давать ложные срабатывания (независимо, есть вагоны перед камерой или их нет). В случае, если на кадре даже человек не способен обнаружить сцепку в силу различных непреодолимых обстоятельств (нет освещения, сцепка абсолютно сливается с фоном, кадр и/или объект засвечены и т. п.), алгоритм так же её не учитывает.

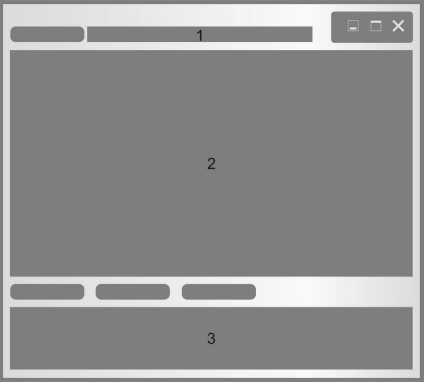
Функции модуля:

Демонстрационный вариант модуля должен позволять выбрать и отобразить произвольный ролик в формате .avi и визуально отображать «сопровождение» сцепки при ее наличии на видео (выделять сцепку прямоугольником на каждом кадре).

На интерфейсе модуля должен быть расположен счётчик, который будет увеличиваться по мере появления новой сцепки в кадре.

Требования к интерфейсу:

Пример возможного интерфейса представлен на рисунке ниже.



Минимальная функциональность:

1. Загрузка видеофайла с возможностью выбора из локальной папки, файл формата – .avi (1);
2. проигрывание выбранного видеофайла (2) с возможность запуска/остановки/паузы (соотв. кнопки);
3. логирование результатов обнаружения сцепок для отображаемого видеофайла в текстовый файл. Например, в каждом сообщении лога информация:

* номер кадра (или время кадра от начала ролика);
* координаты ограничивающего сцепку многоугольника или…
* координаты центра сцепки + ширина и высота объекта.

1. вывод лога в журнал (3);
2. необходимо иметь возможность обрабатывать набор изображений из указанной папки с целью тестирования. Входные данные – .jpg – или .bmp – или .png-изображения. На выходе – файл/файлы, содержащие для каждого изображения координаты ограничивающего сцепку многоугольника. Может быть реализовано в виде отдельного консольного приложения или скрипта для разового запуска.