

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий
(наименование института полностью)

Кафедра «Прикладная математика и информатика»
(наименование)

02.03.03 Математическое обеспечение и администрирование информационных систем
(код и наименование направления подготовки, специальности)

Технология программирования
(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему «Применение метода «Гистограмма ориентированных градиентов» для разработки ПО блока диагностики светотехники автомобиля»

Студент

Е.Л. Ромашов
(И.О. Фамилия)

(личная подпись)

Руководитель

С.В. Митин
(ученая степень, звание, И.О. Фамилия)

Консультант

О.А. Головач
(ученая степень, звание, И.О. Фамилия)

Тольятти 2020

Аннотация

Название бакалаврской работы – «Применение метода «Гистограмма ориентированных градиентов» для разработки ПО блока диагностики светотехники автомобиля».

Объект исследования – применение решений на базе «Гистограмма ориентированных градиентов». Поиск подходящих алгоритмов распознавания объекта и его состояния на изображении и разработка соответствующего ПО.

Предмет исследования – способы выделения информации в видеопотоке с применением технологии ИИ.

Цель исследования – на базе выбранной методики выполнить разработку ПО для прототипа системы диагностики состояния блока фары автомобиля.

Для достижения цели решаются следующие задачи:

- проанализировать существующие способы алгоритмов распознавания объектов;
- описать математический аппарат выбранного алгоритма;
- разработать и протестировать систему, на основе построенного математического аппарата и выявленных требований.

В первой главе затрагиваются способы распознавания объектов на изображениях на базе искусственного интеллекта.

Во второй главе описывается математическая модель выбранного для решения задачи метода и причины, по которым она была выбрана. Описывается математический аппарат алгоритма решающего поставленную задачу и его реализация.

В третьей главе предоставляются результаты тестирования разработанного алгоритма на основе построенного математического аппарата и выявленных требований. Производится корректировка системы и описываются возможности по её улучшению

Бакалаврская работа выполнена на 46 страницах, состоит из введения, трёх глав, включающих 31 изображений, 7 таблиц и 11 формул, заключения, списка используемых источников.

Abstract

The title of the bachelor's work is "Application of the method" Histogram of oriented gradients "for the development of SOFTWARE for diagnostics of vehicle lighting engineering".

The object of research is the application of solutions based on "Histogram of oriented gradients". Search for suitable algorithms for recognizing an object and its state in the image and develop the appropriate SOFTWARE.

The subject of the research is ways to allocate information in a video stream using AI technology.

The purpose of the study is to develop SOFTWARE for a prototype of a system for diagnosing the condition of the car's headlight unit on the basis of the chosen method.

To achieve this goal, the following tasks are performed:

- analyze existing methods of object recognition algorithms;
- describe the mathematical apparatus of the selected algorithm;
- develop and test the system based on the built mathematical apparatus and identified requirements.

The first Chapter deals with ways to recognize objects in images based on artificial intelligence.

The second Chapter describes the mathematical model of the method chosen for solving the problem and the reasons why it was chosen. The mathematical apparatus of the algorithm that solves the problem and its implementation are described.

The graduation work is done on 46 pages, consists of an introduction, three chapters, including 31 images, 7 tables and 11 formulas, a conclusion, a list of sources used.

Оглавление

Введение.....	5
Глава 1 АНАЛИЗ МЕТОДОВ И ПОДХОДОВ РАСПОЗНАВАНИЯ ОБЪЕКТОВ.....	7
1.1 Анализ задачи распознавания объектов на изображении.....	7
1.2 Метод гибкого сравнения на графах.....	8
1.3 Свёрточные нейронные сети.....	10
1.3.1 Общая информация об нейронных сетях	10
1.3.2 Архитектура сверточной нейронной сети.....	15
1.4 Гистограмма направленных градиентов и метод опорных векторов.....	17
1.5 Обоснование выбранного метода.....	19
Глава 2 РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ.....	21
2.1 Выбор используемого программного обеспечения.....	21
2.2 Принцип работы метода НОГ	21
2.3 Подготовка обучающей выборки	26
Глава 3 ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	29
3.1 Обучение детекторов	29
3.2 Распознавание горячей фары.....	36
3.3 Оценка разрешения камеры и времени затрачиваемое на решение задачи.....	42
Заключение	43
Список используемой литературы	45

Введение

В современных реалиях идет активное развитие технологий, и автоматизация приходит в любой процесс жизни человека. Это уже не является чем-либо удивительным то, что описывалось фантастами в далеком прошлом уже является реальным. Сегодня системы использующие машинное зрение и искусственный интеллект на базе нейронных сетей, это современный тренд.

В данном проекте ставится задача применения «Гистограммы направленных градиентов» для распознавания и определения исправности светотехники, а именно блока-фары автомобиля. Блок-фара представляет конструктив, в который интегрировано несколько источников света.

По сигналу от оператора происходит активация системы, в систему приходит сигнал начала теста, по этому сигналу требуется определить наличие блока-фары. И по дополнительным сигналам различить состояние блока-фары. Предполагается что система изначально определит наличие объекта и после определит дальнейший ход действий.

Принцип действий при обнаружении объекта должен быть примерно следующим. По сигналу система запускается и определяет наличие объекта после чего просит оператора включить ближний свет. Оператор в свою очередь выполняет действие и нажимает на своём пульту кнопку – горит. А система в свою очередь подтверждает или не подтверждает состояние блока фары. Таким образом происходит поочередно проверка всех ламп. На вход поступает сигнал о начале проверки, а на выход поступает сигнал верно или не верно и с какой точностью система это определила.

В таком порядке должна работать конечное программное обеспечение. Чтобы реализовать данный модуль необходимо понять, как устроены методы по обнаружению объектов. Рассмотреть основные принципы реализации нейронных сетей и определить наиболее подходящий метод для решения данной задачи.

Не менее важным фактором будет выбор оборудования, с помощью которого будет происходить получение информации об объекте. Сюда входит камера и способ её взаимодействия с модулем, то есть получение данных об объекте.

Цель выпускной квалификационной работы (ВКР) - исследование методов локализации и классификации объектов на изображении и разработка универсального алгоритма для решения практико-ориентированных задач.

Объект ВКР - распознавание блока фары и соответствующих её состояний.

Предмет ВКР: Способы выделения информации в видеопотоке с применением технологии ИИ.

Основные задачи ВКР:

1. Исследовать методы локализации и классификации объектов на изображении.
2. Разработать алгоритм, решающий практико-ориентированную задачу распознавания объекта и его состояний на изображении;
3. Протестировать разработанное решение и провести анализ полученных результатов.

Глава 1 АНАЛИЗ МЕТОДОВ И ПОДХОДОВ РАСПОЗНАВАНИЯ ОБЪЕКТОВ

1.1 Анализ задачи распознавания объектов на изображении

Поставленная задача по распознаванию изображений выполняется чаще всего в несколько действий. Изначально необходимо определить положение объекта на изображении, а после уже определить к какому классу он относится. И так после работы такой системы можно будет сделать вывод о том присутствует ли объект на изображении и к какому классу он относится или же отсутствует вовсе.

На практике может использоваться только локализация объекта на изображении в тех случаях, когда нас не интересует его классификация. Это может быть применено при детектировании пешеходов, нам не так важно какие это люди. Но в случае поставленной задачи необходимо четко определить состояние фары, то есть провести классификацию, так как это и есть смысл работы.

Рассматривая методы локализации и классификации можно выделить несколько основных категорий:

1. Методы, основанные на машинном обучении.
2. Методы, основанные на определении формы объекта локализуемого изображения и его характерные особенности.
3. Методы, основанные на цветовых признаках локализуемого объекта.

Чтобы определить метод наиболее подходящий для решения поставленной задачи, необходимо рассмотреть наиболее популярные и эффективные способы из описанных категорий. За основы возьмем методы, применяемые в распознавание лиц, так как в поставленной задаче будет только один объект у которого необходимо классифицировать различные состояния.

1.2 Метод гибкого сравнения на графах

Суть данного метода основана на эластичном сопоставлении графов, которые описывают очертания лица. Лица представляются в виде графов со взвешенными вершинами и ребрами.

На этапе распознавания остается один граф неизменным, в то время как остальные подгоняются под форму лица деформируясь с целью наилучшей подгонки к первому графу. В таких системах распознавания графы могут представлять собой как решетку прямоугольного вида или структуру, образованную характерными точками лица.

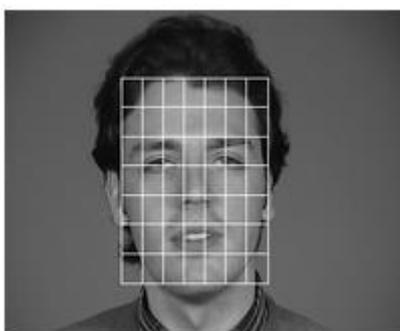


Рисунок 1 – Пример прямоугольной решетки графа

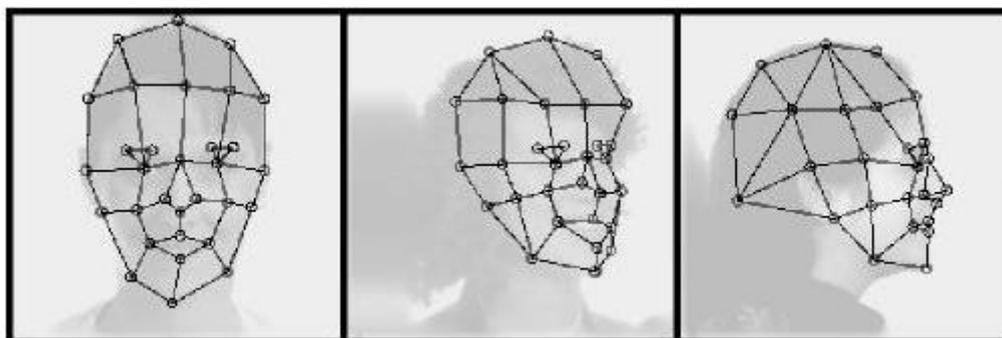


Рисунок 2 – Пример структурного графа

Значения признаков вычисляются в вершинах графа, но обычно используют фильтры Габора или их упорядоченный набор – Габоровский вейвлет (строй Габора), которые вычисляются в некоторой локальной области вершины графа локально, путем свертки значений яркости пикселей с фильтрами Габора.

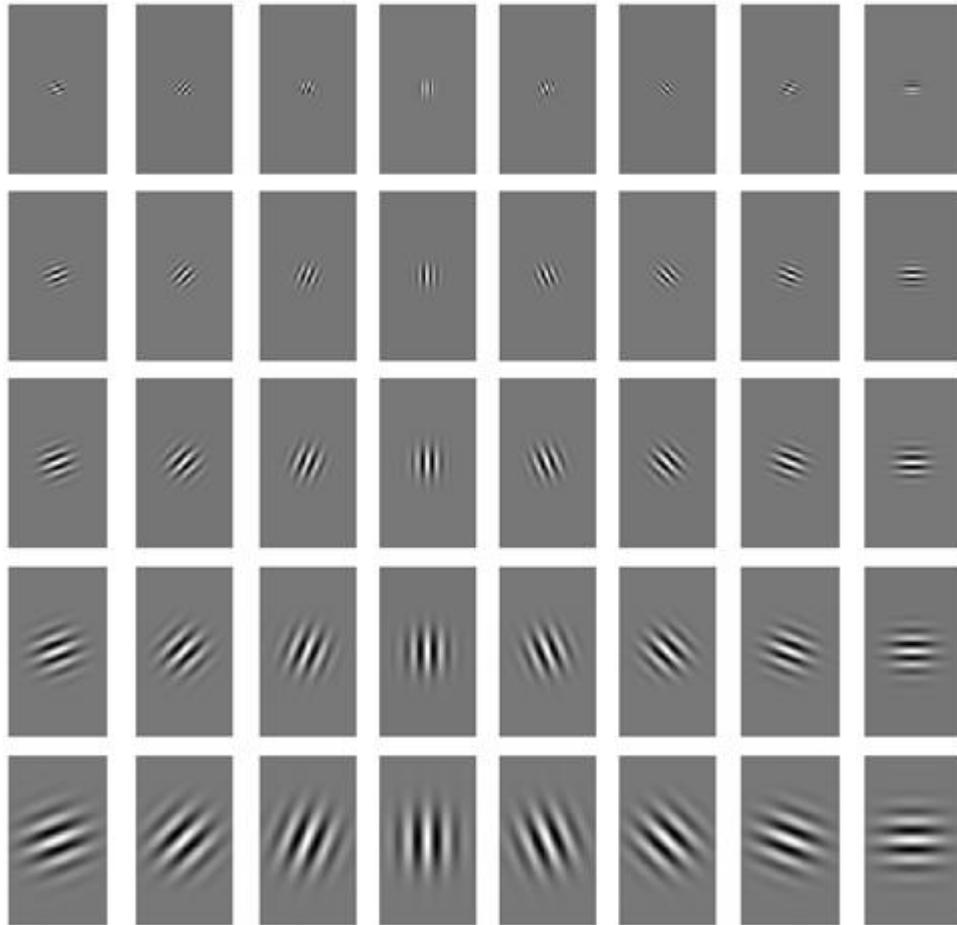


Рисунок 3 – Фильтра Габора

Ребра графа взвешиваются расстояниями между смежными вершинами. Различие между графами вычисляется некоей ценовой функцией деформации, которая учитывает различие между значениями признаков вычисленных в вершинах и степень деформации ребер.

Деформирования графа протекает следующим образом:

1. Смещение вершин на некоторое расстояние в определенных направлениях относительно ей исходного положения
2. Выбор позиции, при которой разница между признаками неизменного графа и деформируемого будет минимальной.
3. Операция выполняется для всех вершин графа, до тех пор, пока не будет получена наименьшая суммарная разница между признаками неизменного графа и деформируемого

В итоге полученное значение ценовой функции деформации и будет являться различием между входным изображением лица и неизменным графом. Процедура деформации должна выполняться для всех изображений, заложенных в базу данных системы. В результате распознавания системы является эталон с наилучшим значением ценовой функции



Рисунок 4 – Пример деформации графа

Точность такого алгоритма указывается как 95-97%-ая эффективность даже при эмоциональных выражениях лиц и изменение ракурса до 15 градусов. Разработчики данной системы ссылались на высокую вычислительную нагрузку данного подхода. Приведен пример, что при наличии 87 неизменных изображений лиц тратится примерно 25 секунд при работе с параллельным ЭВМ с 23 транспьютерами. В других публикациях время работы не указывается или же говорится что оно велико.

1.3 Свёрточные нейронные сети

1.3.1 Общая информация об нейронных сетях

Рассмотрим базовый блок для построения нейронной сети, Искусственный нейрон и принцип его работы. Структура его представлена на рис.5

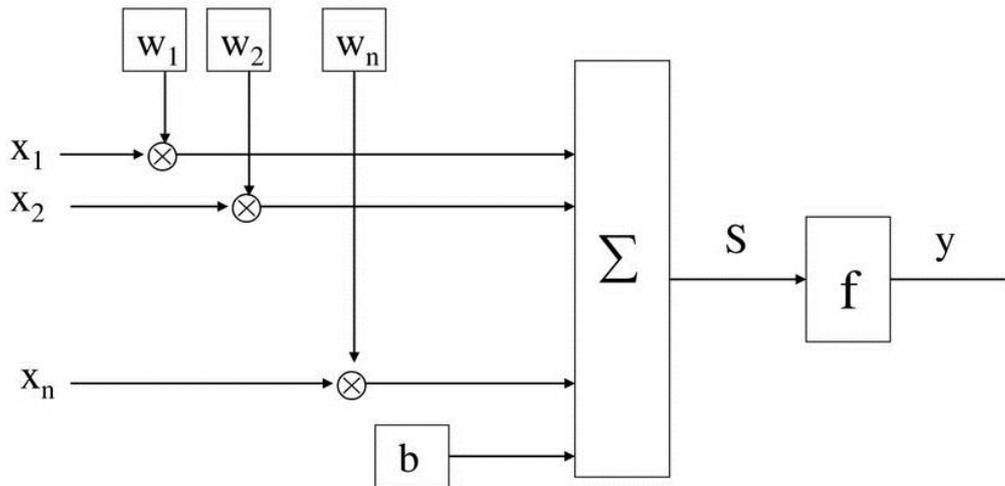


Рисунок 5- Структура искусственного нейрона

Искусственный нейрон состоит из множителей или же синапсов, нелинейного преобразователя и сумматора. Синапсы изображенные перечеркнутым кругом на рисунке, служат связью нейронов между собой и умножают входной сигнал x_i на определенное число. Это число w_i называется весом синапса и характеризует собой силу связи. Последнее сумматор представляет собой сумму всех сигналов, от всех поступающих нейронов к другому нейрону. Нелинейный преобразователь служит для изменения выходного значения сумматора согласно некоторой функции от одного аргумента. Эта функция называется функцией активации служащая для получения значения от нейрона.

Правила преобразования сигналов определяются математической моделью нейрона, которая может быть записана в форме следующих аналитических выражений (1) и (2):

$$s = \sum_{i=1}^n x_i * w_i + b \quad (1)$$

$$y = f(s) \quad (2)$$

Где w_i – вес синапса, x_i – компонент вектора входа или выхода сигнала ($i = \{1, 2, \dots, n\}$), b – значение смещения, s – результат суммирования, n число входов в нейрон, y – выходной сигнал нейрона, f – функция активации представляющая собой некоторое нелинейное преобразование.

Таким образом, отдельно взятый нейрон полностью описывается своей структурой, приведенной на рисунке 1 и математической моделью, приведенной в формулах (1) и (2). Получая вектор значений x , нейрон выдает на выход некоторое число y на выходе. Существует большое количество функций активаций нейрона, рассмотрим только несколько часто встречаемых и используемых.

Первая довольно популярная функция, является сигмоида, описанная формулой (3):

$$y = \frac{1}{1 + e^{-x}} \quad (3)$$

Ниже приведен график сигмоиды отображающий её поведение. Выглядит функции сигмоида гладко, рассмотрим её преимущества.

Во-первых, она не линейна и смотря на рисунок видно, что в диапазоне от -2 до 2 она меняется очень резко. Это означает что при минимальном изменении x приводит к существенному изменению y . Такая особенность показывается что y имеет свойство прижиматься к одному из краёв кривой.

Во-вторых, она не бинарна, что делает активацию аналоговой в отличие от ступенчатых функции, так же для сигмоиды характерен гладкий градиент.

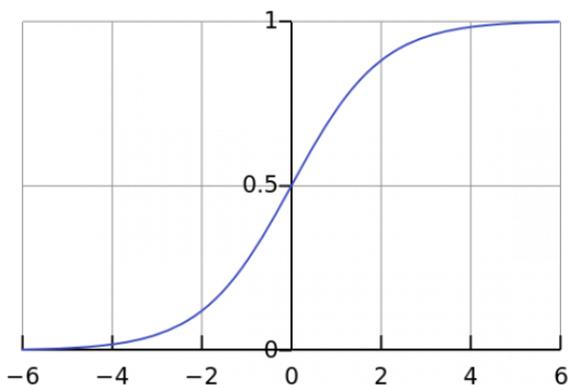


Рисунок 6 – сигмоида

Сигмоида хорошо подходит для решения задач классификации. Функции стремится приблизить значения в пределах от -2 до 2 к одной из сторон кривой. Такое представление помогает находить четкие границы при прогнозе.

На сегодняшний день сигмоид является одной из наиболее частых функций активации в нейронных сетях. Но у этого также есть недостатки, которые заслуживают внимания. Как можно заметить сигмоида при приближении к её концам y имеет свойство слабо реагировать на изменения x . Все это приводит к проблеме градиентного исчезновения, которая в свою очередь влечет за собой незначительное изменение градиента не позволяющее внести существенного изменения. В последствии нейросеть может перестать обучаться или процесс обучения будет протекать медленно.

Следующая функция активации будет гиперболический тангенс, описанный следующей формулой (4):

$$y = \tanh(x) = \frac{2}{1 + e^{-2x}} - 1 \quad (4)$$

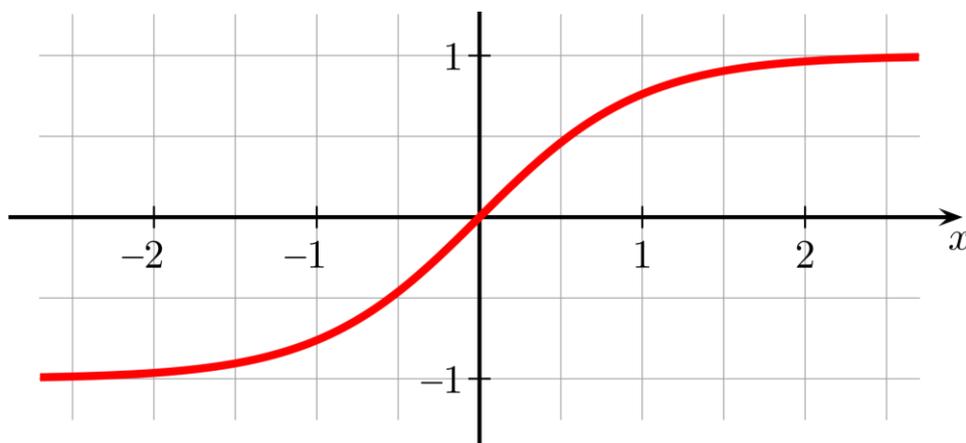


Рисунок 7 – гиперболический тангенс

Функция схожа на сигмоиду и является скорректированной сигмоидной функцией. Поэтому характеристики сигмоиды присущи и данной функции гиперболического тангенса. Небольшое отличие

заключается в градиенте, который у данной функции больше. Все недостатки, имеющиеся у сигмоиды так же свойственны и тангенсу.

Ещё одна функция активации под названием ReLu задаётся следующим уравнением (5):

$$A(x) = \max(0, x) \quad (5)$$

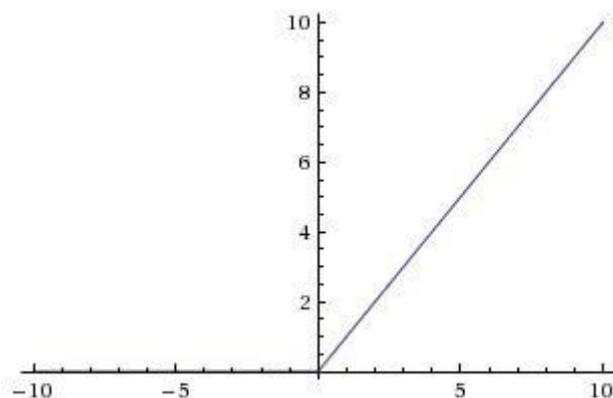


Рисунок 8 – функция ReLu

Такая функция является не линейной и является хорошим аппроксиматором, так как любая функция может быть аппроксимирована данной функцией.

Положительная сторона такой функции является разреженность активации. Это объясняется тем, что при работе с сетью включающую большое количество нейронов при запуске примерно половина активации равны 0, из-за характеристики ReLu возвращать 0 при отрицательном значении y . В такой сети включается меньшее количество нейроном, а значит и работа сети становится легче. Но и это не является однозначным плюсом функции, так как при отрицательных значениях градиент будет равен 0, что повлечет за собой отсутствие коррекции весов во время спуска. Это значит, что пребывающие нейроны в таком состоянии не будут реагировать на изменение ошибки. Решить проблему можно установить значения $y = 0.01x$ для области $x < 0$ что позволит избежать нулевого градиента и постепенно восстанавливать его во время тренировки.

Выбирать функции активации следует из описания поставленной задачи так как различные функции могут быть эффективнее в различных ситуациях. Для определения наиболее эффективной функции можно сделать методом перебора нескольких для определения наиболее точных результатов.

1.3.2 Архитектура сверточной нейронной сети

Сверточная нейронная сеть. По своему названию она дает понять, что действие сети будет происходить по типу свертки изображения большого формата для более эффективной работы с ним. Этот способ несомненно будет полезен при работе с изображением для нашей проблемы, но он зачастую требует больших вычислительных мощностей.

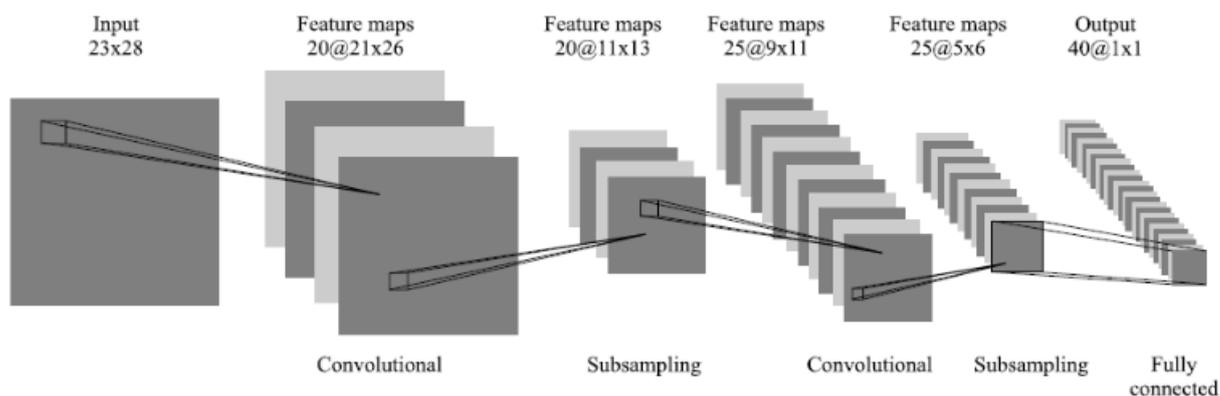


Рисунок 9 – Изображение архитектуры сверточной нейронной сети

Для начала есть некоторое изображение форматом 16×16 пикселей и способные принимать 2 значения. Так же имеется сверточный слой, именуемый фильтром размером 3×3 . Далее происходит следующее, имеющийся фильтр стоит в верхнем левом углу и начинает производить свертку путем перемещения его вправо на 1 пиксель. Сам фильтр должен иметь глубину, соответствующую изображению - это гарантия математической верности. При свертке происходит следующее, начальный квадрат в левом углу на который лег фильтр, перемножается поэлементно с значениями фильтра и суммируется. В итоге получается одно число

символизирующее нахождение фильтра в верхнем левом углу. Такая операция будет продолжаться пока фильтр не пройдет все изображение.

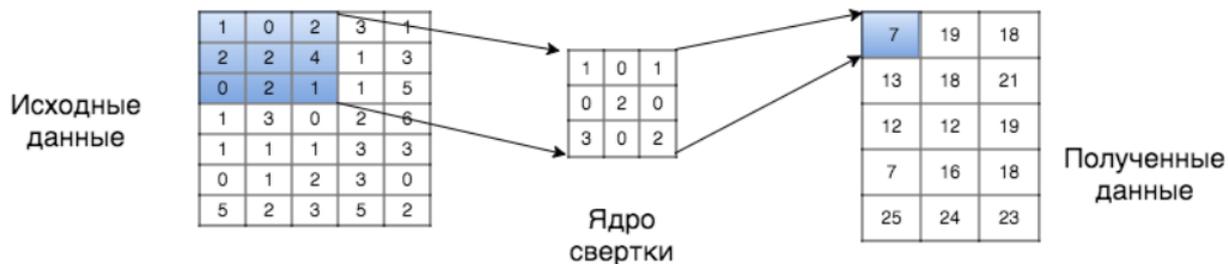


Рисунок 10 – принцип работы свертки

В итоге работы фильтра получится матрица размером 14 x 14 x 1, которую называют функцией активации или картой признаков. Матрица 14 x 14 получается, потому что есть 196 позиций через которые может пройти фильтр 3 x 3 изображение 16 x 16. Все эти 196 чисел преобразуются в матрицу 14 x 14.

У сверточной нейронной сети выделяют несколько присущи им достоинств и недостатков.

Достоинства:

- Присутствует устойчивость к изменению положения и поворотам объекта на изображении
- Меньшее количество параметров по сравнению с полносвязными сетями

Недостатки:

- Для обучения требует длительное время
- Требуется большая обучающая выборка

Не смотря на свойственные архитектуре сверточных сетей недостатки, они по-прежнему остаются наиболее пригодными для решения задачи распознавания объектов на изображении.

1.4 Гистограмма направленных градиентов и метод опорных векторов

Гистограмма направленных градиентов – представляет собой мощный способ преобразования изображения в характеристический вектор (дескриптор). Такой метод является эффективным для распознавания, объектов которым не свойственно существенное изменение в форме. Применять данный метод при перемещении или повороте объекта не рекомендуется, так как он плохо устойчив к таким смещениям. Но это не мешает в решения поставленной задачи, так как фара и камера по итогу будут находится всегда в одинаковом положении.

Билл Триггс и Навнит Далал впервые описали в своей работе гистограмму направленных градиентов в июне 2005 года.

Название гистограмма направленных градиентов является переводом с английского Histogram of Oriented Gradients (HOG).

Метод HOG основан на суммировании направлений градиента в локальных областях изображения. Особенность данного метода является то, что при нахождении гистограммы значения вычисляются равномерно на сетке распределенных ячеек и применяется нормализация перекрывающаяся локальный контраст для увеличения точности.

В описанной работе Билла и Навнита рассматривался алгоритм нахождения пешеходов на изображении, но в последствии алгоритм получил широкое распространение и в других сферах обнаружения объектов в видеопотоке.

Главная идея состоит в том, что объект можно описать и использованием градиентов, которые показывают направление интенсивности контуров. Чтобы получить данные дескрипторы необходимо разбить изображение на ячейки и рассчитать для каждой из них HOG. После соединив полученные диаграммы получается дескриптор, который уже можно использовать для самой локализации объекта по которому он был построен. Чтобы получить наибольшую точность лучше всего использовать не одно изображение, а несколько с объектом на них который необходимо

определить. Так же чтобы увеличить точность было принято решение провести нормализацию по контрасту, что приводит к инвариантности освещения и прочих шумов.

Совместно с HOG детектором используют и метод опорных векторов (SVM - Support Vector Machine). Это достаточно простой классификатор, работающий как в двумерном пространстве, так и на гиперплоскостях.

С двумерными плоскостями все просто, на плоскости лежит некоторое количество точек, а между точек проведена прямая. Разделенные точки прямой относятся к одному из двух классов разделенных проведенной прямой.

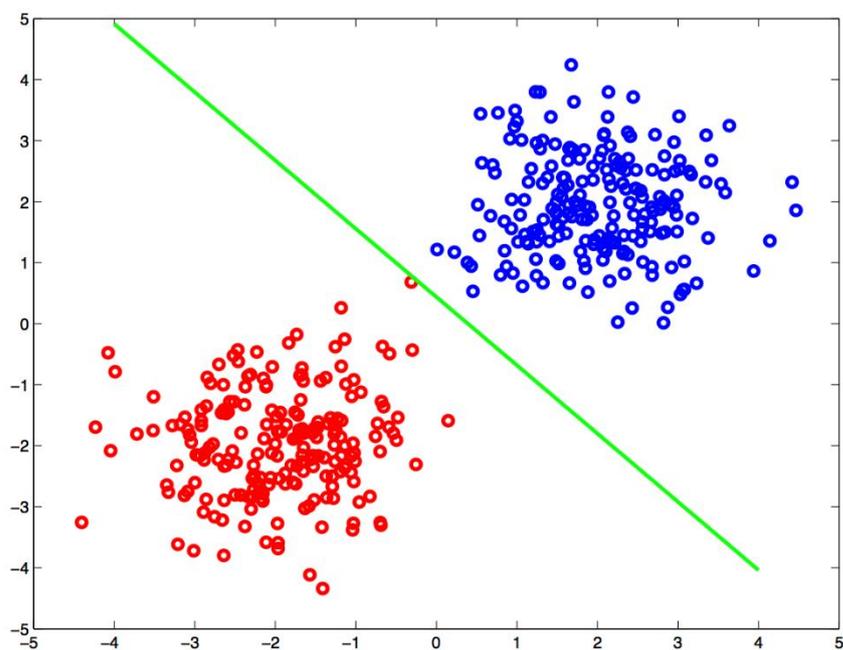


Рисунок 11 – Пример классификации SVM

При решении задачи классификации HOG дескрипторов двумерная плоскость не подходит, потому что дескрипторы уже имеют большую размерность. Для решения таких задач используют разделение гиперплоскостями.

Сама гиперплоскость представляет из себя пространство меньшее на единицу чем исходное. Поэтому для решения задачи, например, имеющую три измерения, гиперплоскость будет обычная двумерная плоскость.

Чтобы решить задачу классификации будет нужно провести несколько гиперплоскостей и выбрать среди них такую, которая будет максимально удалена от обоих классов.

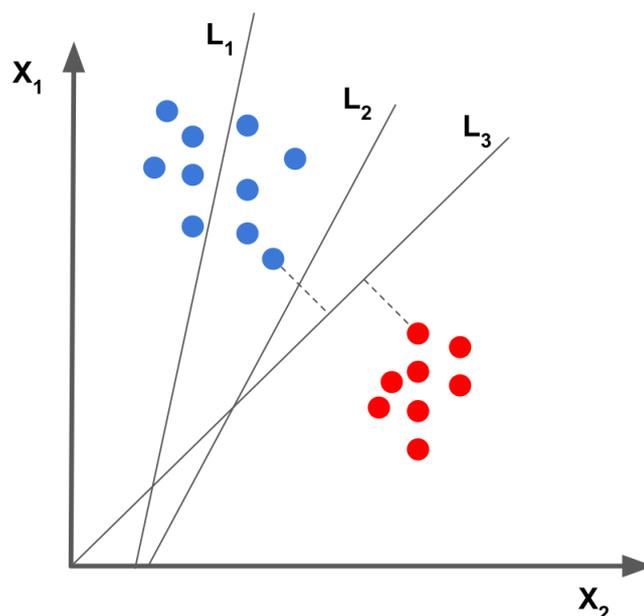


Рисунок 12 – Пример поиска опорных векторов с помощью разделения гиперплоскостями

По итогу вектора, находящиеся ближе всего к гиперплоскости, которая их разделяет, будут опорными векторами. На рисунке 11 вектора — это точки от которых проведены пунктирные линии.

1.5 Обоснование выбранного метода

Рассмотрев несколько способов детектирования объектов на изображении было решено остановиться на методе HOG. Он не требует длительного обучения в отличие от сверточных сетей и большой обучающей выборки. Чтобы решить задачу потребуются обучить несколько дескрипторов, которые будут различать изначально присутствие объекта, а после уже определять в каком состоянии он находится. Так как классифицировать необходимо не много состояний такой метод хорошо подойдет для решения поставленной задачи.

Не маловажным фактором является и скорость работы таких дескрипторов, и высокая точность исходя из полученной небольшой выборки. Сравнивая метод HOG с применением сверточных нейронных сетей в скорости останется преимущество за дескрипторами, как в обучении, так и в обнаружении, хотя и последнее может отличаться незначительно.

Так же не маловажным фактором в обучении дескрипторов является наличие большого количества инструментов, которые помогут упростит разработку программного обеспечения. Одна из реализаций HOG детектора присутствует в библиотеке `dlib` написанная на языке C++, но поддерживающая вызов функции в языке Python. В ней содержится необходимый функционал для обучения и проведения тестирования полученных детекторов.

Вывод к главе 1

Рассмотренный в данной главе выпускной квалификационной работы материал описывает некоторые существующие способы локализации и классификации объектов на изображениях, и позволяет сделать следующие выводы:

1. Существует множество различных классификаторов объектов для различных задач.
2. Описаны некоторые существующие способы для локализации и классификации объектов.
3. Было принято решение для использования метода HOG с классификатором SVM для решения поставленной задачи.

Глава 2 РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

2.1 Выбор используемого программного обеспечения

Для реализации поставленной задачи необходимо понять, что будет более эффективно в рамках её реализации. В настоящее время существует большое количество нейронных сетей для различных видов задач, но для поставленной задачи было решено использовать гистограмму ориентированных градиентов (HOG - Histograms of Oriented Gradients).

В области применения метода HOG есть несколько библиотек на языке Python, это OpenCV и dlib, они помогут при работе с видеопотоком и последующими обучением детекторов.

Среда разработки имеет не такое сильное значения, но для удобства был использован Eclipse с подключенным модулем PyDev для языка Python.

Итак, с основными инструментами все решено и можно приступить к реализации. Для этого в Python существует большое количество библиотек, чтобы реализовать задачу по обнаружению объекта была выбрана библиотека dlib. В ней содержатся различные функции и инструменты для работы с изображением.

2.2 Принцип работы метода HOG

В основе сети лежит гистограмма ориентированных градиентов, ранее она применялась для обнаружения пешеходов. Исследователи Навнит Далал и Билл Триггс впервые описали свою в своей работе гистограмму направленных градиентов в 2005 году. В последствие получила широкое распространение и в других областях из-за своей эффективности распознавания. Основой такого алгоритма является что вид и форма объекта могут быть описаны распределением градиентов интенсивности.

Подход данного метода основан на подсчете градиента яркости в локальных частях изображения. Из сказанного следует, что распределение градиентов яркости на любых частях изображения, дает представление о внешнем виде объекта, который находится на данном участке. Работа такого

метода происходит следующим образом, входное изображение делится равномерно сеткой на одинаковые области, после для каждой такой области вычисляется гистограмма направленных градиентов яркости. Для обеспечения неизменности по отношению к освещению гистограммы проходят нормализацию по контрасту с мерой яркости вычисленной по большему фрагменту.

Полученная сумма нормализованных гистограмм будет представлять собой дескриптор объекта. Дескрипторы полученные в ходе таких преобразований инвариантны к освещению, геометрическим и фотометрическим преобразованиям, не относятся в исключения только сам объект.

Заключительным шагом будет классификация дескрипторов при помощи обучения с учителем.

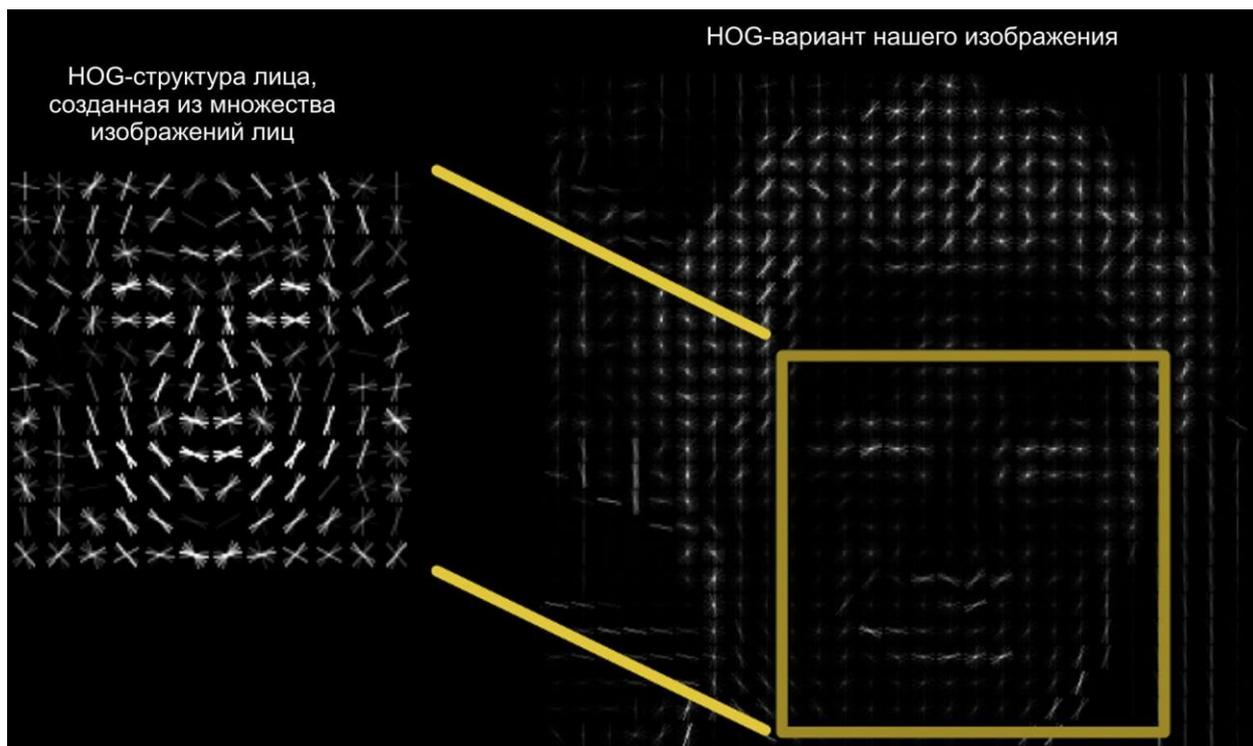


Рисунок 13 – Пример НОВ детектора

На рисунке приведен пример преобразования изображения изначально в гистограмму ориентированных графов, а после путем обучения на некотором количестве изображений лица получается и сам детектор, который

похож на лицо человека. Таким образом можно определять и другие предметы в различных областях предварительно обучив детектор.

Рассмотрим принцип работы такой сети постепенно от входа изображения до получения результата. На приведенном алгоритме, рисунок 14, видно последовательность преобразований, которые осуществляет сеть при работе с изображением. Первым пунктом является нормализация гаммы и цвета, то есть изображение становится черно-белым так как для определения объекта цвет не имеет значения. Так же это может повысить производительность, но очень незначительно.



Рисунок 14 – Алгоритм работы сети

Следующий шаг будет вычисление градиента в ячейках, на которые было разбито изображение. После начинается вычисление градиента следующим образом, берутся все пиксели изображения по порядку и вычисляется на сколько темный является пиксель по сравнению с другими окружающими его пикселями. Целью является провести стрелку в направлении, на котором изображение становится темнее. Самый распространённый способ является применить 1-D разностную маску в горизонтальном и вертикальном направлении. Сделать это можно используя следующие фильтрующие ядра (6) и (7):

$$[-1, 0, 1] \quad (6)$$

$$[-1, 0, 1]^T \quad (7)$$

После применения фильтрующих ядер на изображении получаются величины g_x и g_y с помощью которых находят величину и направление градиента формулам (8) и (9) соответственно:

$$g_r = \sqrt{g_x^2 + g_y^2} \quad (8)$$

$$g_\varphi = \tan^{-1} \frac{g_y}{g_x} \quad (9)$$

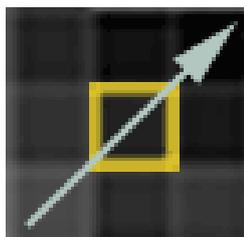


Рисунок 15 – Вычисления градиента для пикселя

При рассмотрении изображенного на картинке видно, что оно темнеет вправо вверх и соответственно туда и будет направлена стрелка. Так будет повторяться для всего изображения и в конечном итоге каждый пиксель будет

заменен стрелкой, эти стрелки и есть градиент, они отображают направление от света к темноте по всей плоскости изображения.

Теперь необходимо провести группировку направлений, то есть создать гистограмму направленных градиентов. Пиксели содержащиеся в локальных ячейках имеют свой взвешенные голоса. Все голоса добавляются в гистограмму учитывая величину и направление градиента. Ячейки на которое было разделено изображение, могут иметь округлую или квадратную формы. Гистограмма может содержать от 0 до 180 каналов при без знаковом градиента и от 0 до 360 при знаковом.

На изображении могут присутствовать различные элементы освещения, для достижения большей устойчивости градиенты нормализуются в своих ячейках. Учитывая изменения в освещении и контрастности, градиент должен быть локально нормализован. Сделать это возможно объединяя ячейки в более крупные блоки, в итоге блоки будут наложены друг на друга как слои, и каждая ячейка внесет свой вклад в дескриптор не один раз.

В своей работе Далал и Триггс применяли 4 способа определения нормировочного множителя L2-норма (10), L2-норма, ограниченная сверху, L1-норма (11), корень из L1-нормы.

$$f = \frac{v}{\sqrt{\|v\|_2^2 + e^2}} \quad (10)$$

$$f = \frac{v}{(\|v\|_1 + e)} \quad (11)$$

В приведенных формулах v – ненормированный вектор, содержащий вес гистограммы, v_k – k-норма этого вектора при $k=1,2$, а e – бесконечно малая константа.

Далал и Триггс опытным путем доказали, что L1 норма менее надежная в отличие от остальных, которые приводят к стабильно хорошим результатам.

В результате работы алгоритма входное изображение преобразуется в характеристический вектор. Длина вектора напрямую зависит от входного изображения и от принятых в ходе вычислений размеров блоков и ячеек.

По итогу всех преобразований необходимо провести обучение с учителем, и был применен метод опорных векторов (SVM). Все характеристические вектора представлены точками в пространстве n . Вектора или же точки могут принадлежать к одному из двух классов. Общая задача состоит в нахождении гиперплоскости с размером $n - 1$, делящей все точки на два класса.

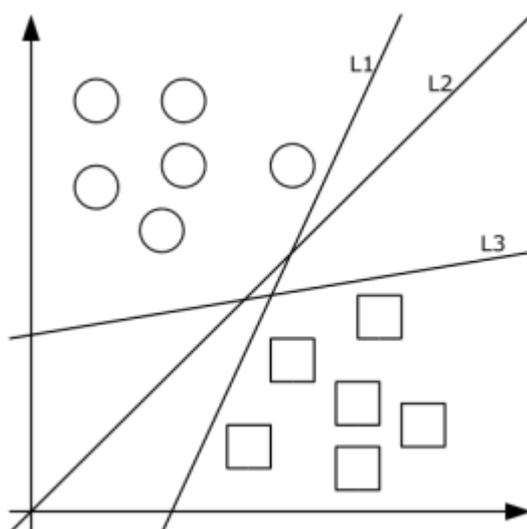


Рисунок 16 – Пример классификации с помощью SVM

Чтобы обучить такой алгоритм не нужна большая выборка изображений. В нашем случае необходимо определить один объект и его состояния после, поэтому потребуется несколько детекторов для различных состояний и положений объекта. В первую очередь нужно понять есть ли сам объект на изображении, а после уже проверять в каком его состоянии он находится.

2.3 Подготовка обучающей выборки

Для обучения детекторов необходимо подготовить данные для подачи их на вход. Изначальные данные состоят из видеоряда, что не совсем подходит для входных данных, поэтому исходный видеоряд был нарезан на кадры и в общей сумме получилось 1038 изображений. В подготовке таких кадров помогла библиотека `OpenCV`, в которой присутствует инструмент для кадрирования видео. Но это тоже нельзя отдавать на вход для обучения, так как тут содержатся кадры, на которых изображение может быть сильно размыто, что мешает корректному обучению детекторов. Поэтому были отобраны наиболее четкие изображения, на которых изображен объект. После отбора все изображения были помещены в отдельный каталог для дальнейшей обработки.

Теперь имея некоторую выборку можно начинать дальнейшую обработку. На этом этапе необходимо обозначить на изображении нахождение объекта выделяя его. В библиотеку `dlib` для этого предоставляется инструмент `imglab`. Функционал данного инструмента помогает произвести дальнейшую обработку путем выделения объекта.

Изображения, отобранные ранее помещаются в один каталог для `imglab`. Теперь запуская данный инструмент необходимо изначально создать файл формата `xml` содержащий все отобранные изображения для обучения. После чего открыть его используя тот же инструмент и начинать выделять объект на всех присутствующих изображениях.



Рисунок 17 – пример обработанного изображения

Как представлено на рисунке выше необходимо отметить объект на всех изображениях перед тем как использовать их для обучения.

Теперь, когда обучающие данные отобраны и подготовлены можно их подавать на обучение дескрипторов.

Вывод к главе 2

Материал приведенный в данной главе рассматривает следующие пункты:

1. Было выбрано используемое программное обеспечение и необходимые инструменты.
2. Рассмотрена архитектура математическая модель и принцип работы алгоритма HOG.
3. Подготовлена выборка данных для обучения детекторов.

Исходя из рассмотренного алгоритма будет реализована программное обеспечение для обучения и детектирования объекта. Также классификации его состояний.

Глава 3 ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

3.1 Обучение детекторов

Чтобы обучить наши детекторы потребуется ранее подготовленная выборка для обучения. В ней содержатся 15 изображений с фарой выделенных как показано на рисунке 17. Это необходимо для обучения детектора, который в последствие будет использоваться в обнаружение объекта.

Для большей эффективности необходимо и определять объект в других положениях, например, если камера будет наклонена вправо, то и объект сместится. Это непосредственно минус данного метода, но незначительно усложнит процесс обучения т.к. может понадобится несколько детекторов.

Обучение детектора продолжается до тех пор пока на обучающей выборке не будет достигнута точность в единицу, то есть на всех изображениях объект был обнаружен. Для тестирования детектора так же подготовлены выборка, на которой детектор уже не обучался, но имеющая небольшие различия с обучающей для проверки работоспособности детектора.



Рисунок 18 – Полученный детектор после обучения

На приведенном рисунке 18 отображен полученный детектор после обучения с алгоритмом НОГ. Для достоверности представим нормализованное изображение при помощи НОГ фильтра, и сравним с предоставленным детектором. Если учитывать общее очертание они должны иметь схожий вид.

Histogram of Oriented Gradients

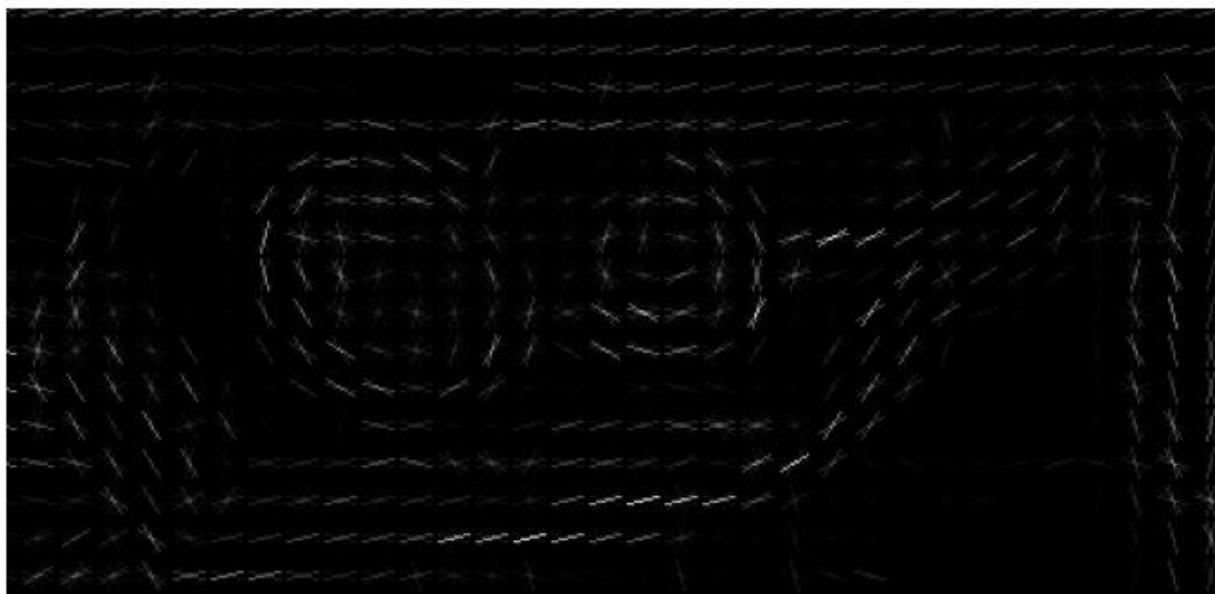


Рисунок 19 – Нормированное изображение

Сравнивая полученный детектор на рисунке 18 и изображение на рисунке 19 видно, что присутствуют отдаленные очертания объекта – фары. Значит алгоритм работает корректно и на выходе получают нужные дескрипторы.

Input image



Рисунок 20 – Объект из обучающей выборки

Сравнивая рисунки 18 и 20 видно, что полученный детектор отдаленно похож на объект, который необходимо обнаружить. Исходя из этого можно сделать вывод что даже на небольшой выборке для обучения полученный детектор сможет обнаружить объект, находящийся в схожем положении на других изображениях.

Для достоверности проверим полученный детектор на нескольких изображениях и сравним полученные результаты.

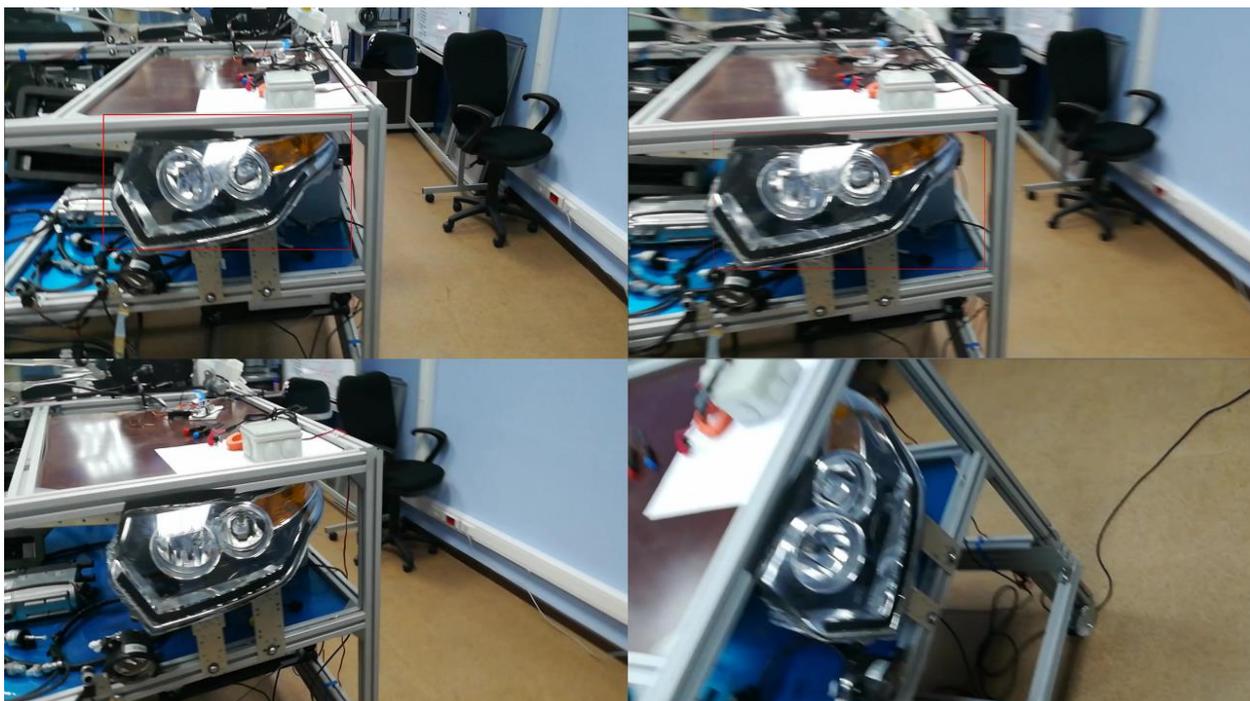


Рисунок 21 – Тестовые изображения

На рисунке 21 приведены примеры тестовых изображений подаваемых на проверку детектору. Можно заметить что на первых двух из них детектор смог распознать объект, а на последующих двух нет. Первой причиной не обнаружения послужила маленькая выборка содержащая в себе изображения практически одного типа. Но этого хватит чтобы понимать что обученный детектор функционирует нормально и способен распознавать объект.

Ниже приведена таблица с номерами изображений и точностью обнаружения для каждого из них. Нумерация изображений на рисунке 20 идет слева направо сверху вниз.

Таблица 1 – Сравнения точности обнаружения первого обученного детектора

Номер изображения	Точность обнаружения
Первое	0,93
Второе	0,391
Третье	Менее 0,001
Четвертое	Менее 0,001

Теперь для большей точности необходимо увеличить выборку для обучения чтобы в неё попадали изображения объекта с незначительными отклонениями и различными положениями на изображении. для наглядности используем аналогичные примеры изображений для тестирования обученного детектора.

В новой выборке содержится 54 изображения, что уже значительно повлияло на время обучения которое увеличилось в несколько раз.. Но это никак не влияет на работу детектора по времени, так как главное чтобы детектор выдавал верный результат.

После обучения вид детектора приобрел следующие очертания на рисунке 22. Сравнивая с первым детектором на рисунке 18, можно заметить что присутствуют отличия по внешнему виду. Это связано с добавленными изображениями в которых объект находится под наклоном. Теперь как и предыдущий детектор проверие новый полученный на уже ранее использованных изображениях.



Рисунок 22 – Полученный детектор после второго обучения

Приведенная таблица отображает данные в соответствии обнаружения для второго детектора по аналогичным данным.

Таблица 2 – Сравнения точности обнаружения первого обученного детектора после изменения обучающей выборки

Номер изображения	Точность обнаружения
Первое	0.96
Второе	0.89
Третье	0.935
Четвертое	Менее 0.001

Исходя из полученных данных можно сделать следующие выводы, что эффективность обучаемого детектора напрямую зависит от количества изображений, находящихся в обучающей выборке. Так же туда можно и отнести качество изображения, которое играет не маловажный фактор.

Сравнивая данных таблицы 1 и 2 отчетливо видно разницу в точности на первых трех изображениях. На первом и втором изображении она составляет единицу и только на втором немного отклонилась от этого показателя, это может быть связано с небольшой размытостью третьего. Последнее же изображения так и не было распознано потому что в выборке не было такого сильного отклонения объекта и использовалось как показатель что необходимо обучения и для наклонных положений объекта.

Аналогичные действия сделаем и для второго детектора, который будет отвечать за распознания объекта в другом положении, в данном случае это будет поворот камеры вправо. Для тестирования используем также четыре изображения и полученные результаты выведем в таблицу для наглядности.

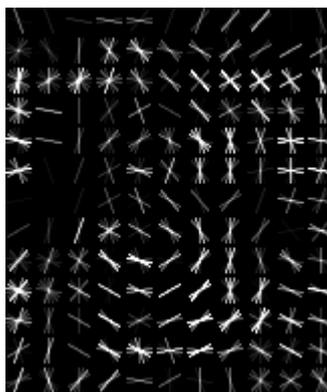


Рисунок 23 – Второй обученный детектор

При обучении второго детектора использовалась сразу большая выборка исходя из прошлого опыта с первым детектором. Изначально трудно рассмотреть очертания объекта, но можно заметить, что есть схожесть с приведенным на рисунке 22. Но тут были гораздо более разнообразные обучающие данные, потому что объект в каждом последующем кадре смещался.

Проверяя полученный детектор на новых тестовых изображениях результат сразу оказался хорошим, так как на всех изображениях удалось распознать объект. Отчасти на это несомненно повлияла отобранная выборка в этот раз она содержала сразу все варианты расположения объекта.

Позже полученные детекторы можно запускать одновременно для детектирования объекта при любом положении, для которого они обучались. Таким образом будет представлена некая нечеткость в работе обнаружения объекта. Далее для этого будут обучены другие детекторы, которые уже будут определять состояние фары, то есть какой свет горит в зависимости от успешности обнаружения того или иного детектора.



Рисунок 24 – Тестовые изображения для второго детектора

Все результаты по точности работы второго детектора представлены в таблице 3.

Таблица 3 – Сравнения точности обнаружения второго обученного детектора

Номер изображения	Точность обнаружения
Первое	0.871
Второе	0.583
Третье	0.93
Четвертое	0.501

В приведенной таблице 3 представлены результаты точности с которой второй детектор распознал объект. Только на двух изображениях точность ниже, вполне возможно, что на это повлияла четкость на них самого объекта, так как на первом и третьем изображении объект отображается четко.

3.2 Распознавание горячей фары

После распознавания самого объекта необходимо понять какой свет горит на самой фаре. Для этого рассмотрим, когда горит ближний и дальний свет по отдельности в совместно. Чтобы распознать эти состояния необходимо обучить ещё три детектора которые свою очередь будут представлять некоторую нечеткость сети, так как исходя из сравнения соответствия обнаружения будет определяться в каком из состояний сейчас находится фара.

Чтобы обучить детекторы для определения какой свет горит у фары. Допустим, что первый детектор будет определять, когда горит ближний свет, второй, когда горит дальний свет и третий, когда горят только ходовые огни. Не стоит забывать, что в первых двух случаях всегда включены ходовые огни, а при включенном дальнем свете всегда включен ближний свет. В таком случае есть вероятность что несколько детекторов сразу смогут определить объект с некоторой вероятностью. Поэтому будут сравниваться их показатели соответствия определения.

Первый обученный детектор будет служить для определения состояния, когда горит ближний свет соответственно совместно с ходовыми огнями. Подготовим выборку и подадим её на обучение детекторов.

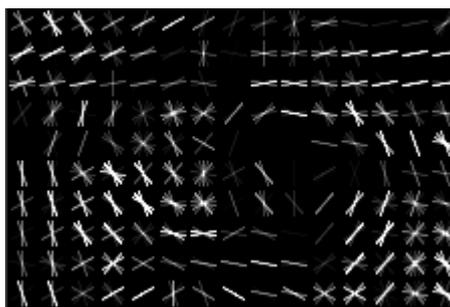


Рисунок 25 – Детектор ближнего света

На рисунке 25 приведен полученный детектор после обучения. Смотря на результат практически невозможно распознать очертания фары, но видно, что появилась черная область, которая соответствует огню ближнего света.

Это проявляется из-за того, что градиент в этой области имеет практически один цвет и поэтому он получается равен 0 в этой области.

Проверим полученный детектор на тестовых изображениях и приведем результаты полученные соответствия каждого изображения.

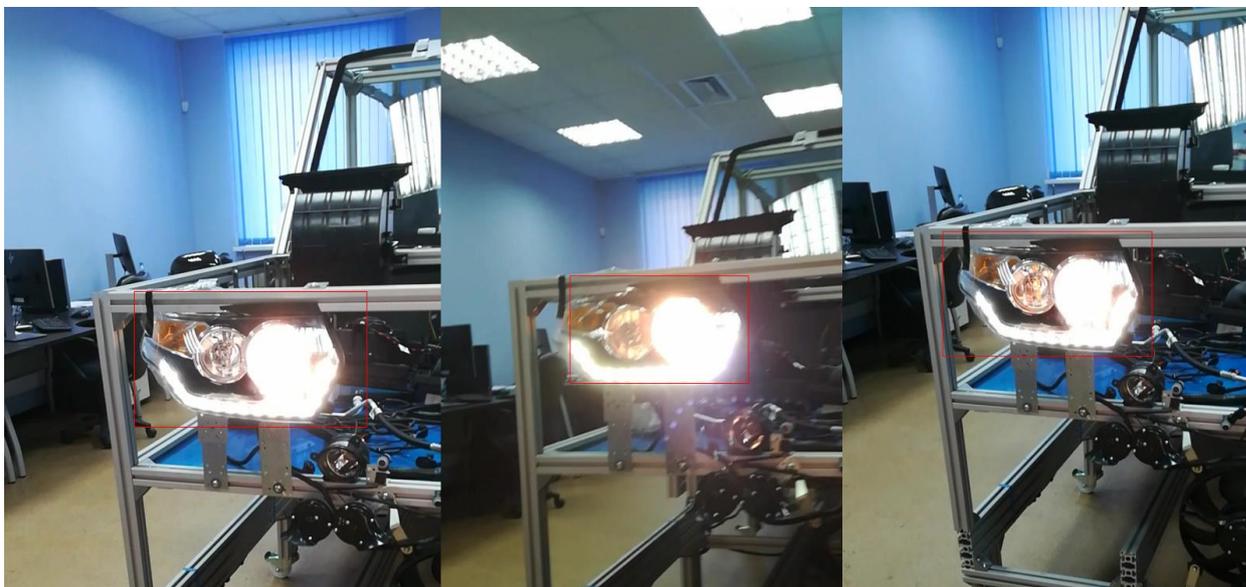


Рисунок 26 – Тестовые изображения для третьего детектора

После проверки на тестовой выборке детектор распознал фару с горящим ближним светом на всех трех тестовых изображениях. Приведем таблицу сравнения точности обнаружения.

Таблица 4 – Сравнения точности обнаружения третьего обученного детектора

Номер изображения	Точность обнаружения
Первое	0,845
Второе	1
Третье	0,745

По приведенным результатам видно, что детектор успешно справился со своей задачей. Значит можно приступать к обучению четвертого по счету детектора.

Для обучения соответственно необходима новая выборка, содержащая в себе состояние, когда горит дальний свет. Но в таком состоянии будет задействован и ближний свет, и ходовые огни. Что может усложнить дальнейшее определение состояния. Сейчас обучим детектор дальнего света и проверим его на тестовых изображениях.



Рисунок 27 – Детектор дальнего света

Приведенный рисунок детектора схож с детектором на рисунке 25. Но, как и в случае с ближним светом на детекторе имеются черные области, каждая из которых отображает горящие лампы ближнего и дальнего света.

Проверим полученный детектор на тестовых изображениях, как и прошлый и представим полученные результаты.

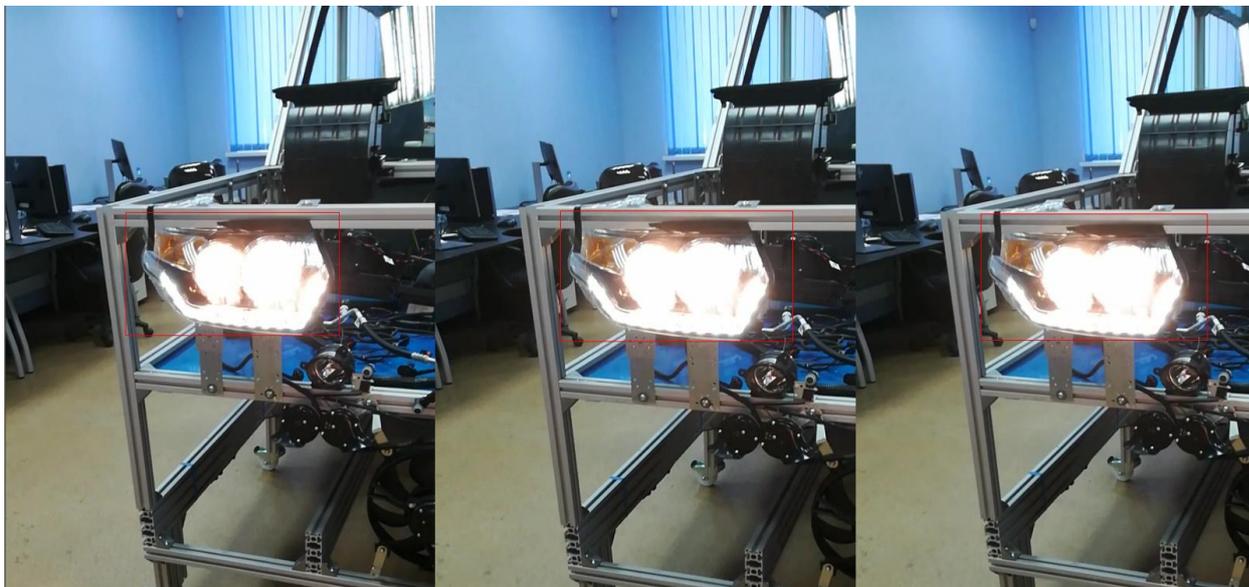


Рисунок 28 – Тестовые изображения для четвертого детектора

Детектор успешно обнаружил объект с горящим дальним светом. Аналогично прошлому приведем таблицу сравнения точности обнаружения.

Таблица 4 – Сравнения точности обнаружения четвертого обученного детектора

Номер изображения	Точность обнаружения
Первое	0,808
Второе	0,924
Третье	0,930

Как и прошлый детектор отлично сработал в обнаружение объекта и показал хорошие результаты на тестовых изображениях. Но теперь необходимо обучить последний детектор для состояния, когда горят только ходовые огни. Как и прежде нужно подготовить выборку для обучения.



Рисунок 29 – Детектор ходовых огней

По полученному детектору видно, что отсутствуют черные пятна в центре так как не горит ближний и дальний свет

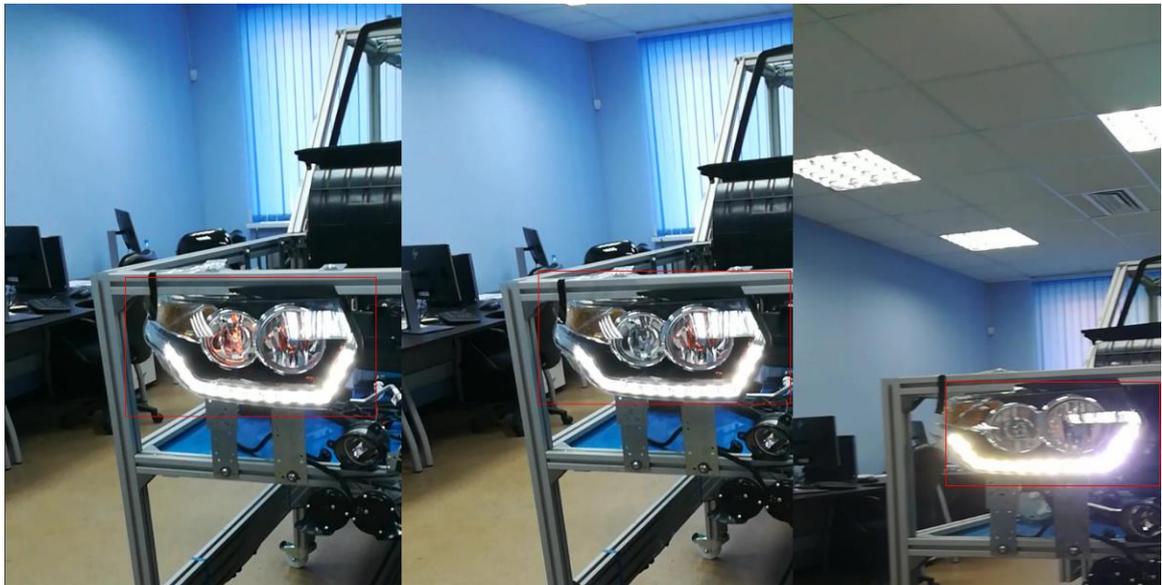


Рисунок 30 – тестовые изображения для пятого детектора

Как и прошлые полученный детектор справился с задачей и распознал на каждом изображении фару с горящими ходовыми огнями. Приведем полученные результаты соответствия распознавания в таблицу.

Таблица 5 – Сравнения точности обнаружения четвертого обученного детектора

Номер изображения	Точность обнаружения
Первое	0,782
Второе	0,730
Третье	0,886

По результатам детектор работает исправно и справляется с поставленной задачей.

После обучения трех детекторов по распознаванию трех различных состояний фары, необходимо проверить их совместную работу для избегания конфликтов между ними. В проверки используем ранее приведенные тестовые изображения на рисунках 26, 28 и 30. Включим все детекторы одновременно для проверки каждого изображения и сравним полученные результаты.

Таблица 6 – Сравнения точности обнаружения состояния фары

Номер изображения	Ближний свет	Дальний свет	Ходовые огни
Первое (Ближний свет)	0,845	0,590	0,547
Второе (Ближний свет)	1	0,643	0,542
Третье (Ближний свет)	0,745	0,179	0,521
Четвертое (Дальний свет)	0,768	0,808	0,591
Пятое (Дальний свет)	0,749	0,924	0,511
Шестое (Дальний свет)	0,776	0,930	0,486
Седьмое (Ходовые огни)	0,509	0,288	0,782
Восьмое (Ходовые огни)	0,499	0,224	0,730
Девятое (Ходовые огни)	0,329	0,096	0,886

В таблице нумерации изображений идут в соответствующем порядке, как и использовались для тестов отдельно взятых детекторов. Сравнивая показатели, которые представлены в таблице получается, что на первых трех изображениях преобладает детектор ближнего света, как и должно быть. На изображениях четыре пять и шесть есть небольшая опасность на четвертом изображении так как уверенность первого детектора очень близка, но не превысила значение истины для неверного результата. На последних трех изображениях ситуация обстоит хорошо и третий детектор показал наибольший результат.

Для наглядности сравним детекторы на большой тестовой выборке. По полученным данным представим гистограмму количества изображений в тестовой выборке и процент детектирования объекта на них.

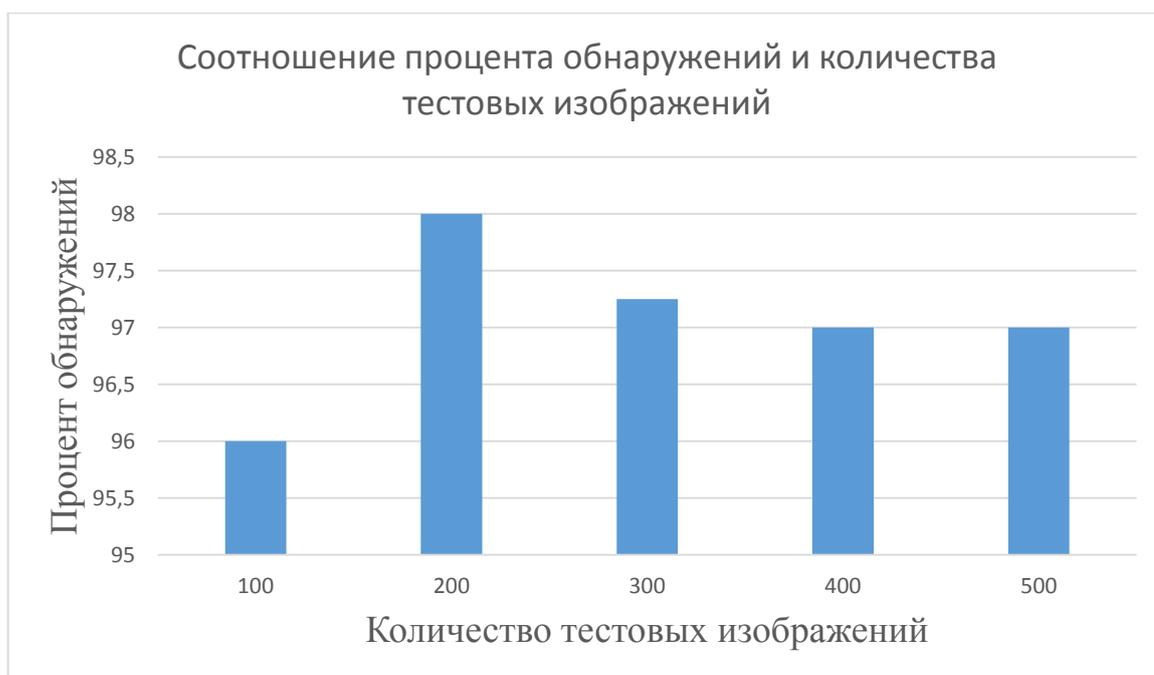


Рисунок 31 – Результаты тестирования

Получается, что все обученные детекторы достаточно корректны для прогнозирования текущего состояния фары учитывая, что ранее они успешно отличали состояния, в котором находится фара можно сделать вывод что в текущем тесте распознавание было корректно. В текущих реалиях процент мог и достигнуть 100% если убрать из выборки сильно смазанные или размытые изображения на которых распознать объект не представляется возможным. Исходя их проведенного теста детекторы работают отлично.

3.3 Оценка разрешения камеры и времени затрачиваемое на решение задачи

В решение поставленной задачи использовалось разрешение камеры 1280x720 пикселей. При обучении сети затрачивалось относительно небольшое время, но учитывать его для определения разрешения нет смысла. Обучение происходит до того, как начнется работа с уже обученными детекторами. Само же полученное изображение достаточно четкое чего вполне хватает для определения объекта на полученных изображениях.

Для примера приведем время, затраченное на обнаружение объекта используя для теста все изображения, рассматриваемые в таблице 6.

Таблица 7 – Время затраченное на обнаружение объекта

Номер изображения	Затраченное время
Первое (Ближний свет)	998 мс
Второе (Ближний свет)	731 мс
Третье (Ближний свет)	730 мс
Четвертое (Дальний свет)	735 мс
Пятое (Дальний свет)	748 мс
Шестое (Дальний свет)	749 мс
Седьмое (Ходовые огни)	753 мс
Восьмое (Ходовые огни)	765 мс
Девятое (Ходовые огни)	755 мс

При тестировании детекторов на изображениях затрачивалось чуть менее одной секунды на каждое отдельно взятое изображение что достаточно быстро и вполне удовлетворяет условиям задачи. Используя такой метод в непосредственно тестировании фары человеком с пульта решено считать, что затраченное время является оптимальным и удовлетворяет решению поставленной задачи.

Заключение

В рамках выпускной квалификационной работы были рассмотрены способы локализации и классификации, были выполнены все задачи и поставленные цели. Разработано ПО для тестирования блока фары требуемое для решения задачи.

В ходе выполнения бакалаврской работы были решены следующие задачи:

1. Изучены некоторые способы локализации и классификации объектов на изображениях и произведен выбор наиболее подходящего способа.
2. Исходя из сделанного выбора был рассмотрен математический аппарат алгоритма применяемого метода, который может в последствии

применятся и в других областях при решении практико-ориентированных задач.

3. Произведено тестирование разработанного программного продукта на реальных данных. В ходе тестирования установлено, что обученные детекторы хорошо справляются с поставленной задачей и определяют к какому состоянию, относится текущее состояние блока фары.

Для обучения и тестирования использовался отснятый видеоматериал на камеру и нарезанный на кадры. Полученные таким образом изображения отбирались для обучения более четкие, для тестов использовались также и смазанные изображения чтобы проверить устойчивость детекторов.

Полученные результаты по применению алгоритма можно считать успешными, во время тестирования не было выявлено никаких ошибок, которые могли бы помешать получению верных результатов.

Разработанное программное решение может использоваться для помощи человеку, проводящему тестирование автомобиля или в будущем может полностью заменить человека на этом этапе тестирования. Так же возможно переобучение алгоритма для решения других практико-ориентированных задач.

Список используемой литературы

1. Алешева Л. Н. Интеллектуальные обучающие системы [Текст] / Л. Н. Алешева // Вестник университета. - 2018. - N 1. - С. 149-155
2. Джонс М. Т. Программирование искусственного интеллекта в приложениях / М. Т. Джонс ; пер. с англ. А. И. Осипов. - М. : ДМК Пресс, 2018. - 311 с.
3. Нейронные сети: практическое применение [Электронный ресурс] Режим доступа: <https://habr.com/ru/post/322392/>
4. Как работает сверточная нейронная сеть: архитектура, примеры, особенности [Электронный ресурс] Режим доступа: <https://neurohive.io/ru/osnovy-data-science/glubokaya-svertochnaja-nejronnaja-set/>
5. Как работает сверточная нейронная сеть: архитектура, примеры, особенности [Электронный ресурс] Режим доступа: <https://neurohive.io/ru/osnovy-data-science/glubokaya-svertochnaja-nejronnaja-set/>
6. Применение методики гистограмм направленных градиентов для классификации дактилоскопических изображений [Электронный ресурс] Режим доступа: <http://elib.bsu.by/bitstream/123456789/179301/1/53-60.pdf>
7. Локализация объектов на изображении методом свёрточных нейронных сетей [Электронный ресурс] Режим доступа: <http://www.azoft.ru/blog/lokalizaciya-obektov-na-izobrazhenii-metodomsvyortochnyx-nejronnyx-setej/>
8. Обучение машины — забавная штука: современное распознавание лиц с глубинным обучением [Электронный ресурс] Режим доступа: <https://habr.com/ru/post/306568/>
9. Как работает сверхглубокая нейронная сеть? [Электронный ресурс] Режим доступа: <https://postnauka.ru/video/66872>
10. Каллан Р. Нейронные сети: краткий справочник / Р. Каллан ; Саутгемптон. ин-т. - М. : Вильямс, 2017. - 279 с.

11. Рассел С. Норвиг П. Искусственный интеллект : современный подход : пер. с англ. / С. Рассел, П. Норвиг ; пер. К. А. Птицын. - 2-е изд. - М. : Вильямс, 2018. - 1407 с.
12. Рашка Себастьян Python и машинное обучение / Рашка Себастьян. - М.: ДМК Пресс, 2017. – 809 с.
13. Плас Джейк Вандер Python для сложных задач. Наука о данных и машинное обучение. Руководство / Плас Джейк Вандер. - М.: Питер, 2018. – 527 с.
14. Jiménez, P. Background Pixel Classification for Motion Detection in Video Image Sequences [Text] / P.Gil Jiménez, S. Maldonado-Bascón, R. Gil-Pita, H. Gómez-Moreno // International Work-Conference on Artificial Neural Networks – 7th International Work-Conference on Artificial and Natural Neural Networks, IWANN 2003 Maó, Menorca, Spain, June 3–6, 2003 Proceedings, Part I: Computational Methods in Neural Modeling. – Springer-Verlag Berlin Heidelberg 2003. – pp. 718-725
15. Navneet Dalal, Bill Triggs. Histograms of Oriented Gradients for Human Detection, 2005. – 8 p
16. Habibi Aghdam. H, Jehani Heravi. E, Guide to Convolutional Neural Networks A Practical Application to Traffic-Sign Detection and Classification, 2017. – 282 p
17. Learning Deep Features for Discriminative Localization [Электронный ресурс] Режим доступа: http://cnnlocalization.csail.mit.edu/Zhou_Learning_Deep_Features_CVPR_2016_paper.pdf
18. Melin, P. Design of Intelligent Systems Based on Fuzzy Logic, Neural Networks and Nature-Inspired Optimization / Patricia Melin, Oscar Castillo, Janusz Kacprzyk. – Springer International Publishing, 2015. – 637 p
19. Yujin Zhang. Image Processing / USA: De Gruyter, 2017. – 294 с

20. James Loy. Neural Network Projects with Python: The ultimate guide to using Python to explore the true power of neural networks through six projects 1st Edition / USA: Packt Publishing, 2019. – 308 с.
21. Ashwin Pajankar. Python 3 Image Processing / USA: BPB Publications, 2019. – 187 с.
22. The history of Neural networks [Электронный ресурс] Режим доступа:
<https://dataconomy.com/2017/04/history-neural-networks/>
23. J. Stallkampa, M. Schlipf, J. Salmena C. Igelb Man vs. Computer: Benchmarking Machine Learning Algorithms for Traffic Sign Recognition, 2012. – 11 p
24. Liu, C.-L., Yin, F., Wang, Q.-F., & Wang, D.-H. ICDAR 2011 chinese handwriting recognition competition. In International Conference on Document Analysis and Recognition, 2011. – 1464 –1469 p
25. Aurelien Geron. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems 2nd Edition / USA: O'Reilly Media, 2019. – 849 с.
26. Histogram of Oriented Gradients [Электронный ресурс] Режим доступа: <https://www.learnopencv.com/histogram-of-oriented-gradients/>

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Гольяттинский государственный университет»

Институт математики, физики и информационных технологий
(наименование института полностью)

Кафедра «Прикладная математика и информатика»
(наименование)

ОТЗЫВ
руководителя о выпускной квалификационной работе

Студент Ромашов Евгений Леонидович
(ФИО полностью)

02.03.03 Математическое обеспечение и администрирования информационных систем
(код и наименование направления подготовки, специальности)

Технология программирования
(направленность (профиль) / специализация)

Тема «Применение метода «Гистограмма ориентированных градиентов» для разработки ПО блока диагностики светотехники автомобиля»

В данной работе представлено программное решение задачи определения состояния блока-фары. Актуально применение современного подхода анализа изображения в новом технологическом оборудовании. Практическая значимость данного решения заключается в том, что с помощью применения доступного оборудования, позволяет без вмешательства, оценить работоспособность блока-фары, что необходимо для решения задач диагностики.

Нужно отметить самостоятельность, проявленную студентом в ходе решения данной задачи. Разработанное ПО для проверки состояния блока-фары использует для анализа изображения метод «Гистограмма ориентированных градиентов». При разработке применялись языки программирования «Python» и «С++», среда разработки Eclipse.

Считаю, что бакалаврская работа Ромашова Е.Л. выполнена на хорошем уровне, имеет практическую значимость, отвечает требованиям, предъявляемым к бакалаврской работе по направлению 02.03.03 «Математическое обеспечение и администрирование информационных систем» и заслуживает оценки «хорошо», а автор присвоения квалификации бакалавра по направлению 02.03.03 «Математическое обеспечение и администрирование информационных систем»

Руководитель,
ст. преподаватель
(ученая степень, звание, должность)

(подпись)

С.В. Митин
(И.О. Фамилия)

« » 20 г.