

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН
Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ»

Кафедра «Электроника и Робототехника»

«ДОПУЩЕН К ЗАЩИТЕ»

Зав. кафедрой

(ученая степень, звание, Ф.И.О.)

« » 201 г.

(подпись)

ДИПЛОМНЫЙ ПРОЕКТ

На тему: Создание модуля распознавания лиц с Raspberry Pi

Специальность 58071600-Приборостроение

Выполнил (а) Белов Максим Михайлович

Группа ПС-15-3

(Ф.И.О.)

Научный руководитель доцент, к.т.н. Шабельников Е.А.

(ученая степень, звание, Ф.И.О.)

Консультанты:

по экономической части:

к.э.н., доцент, Бекишев А.И.

(ученая степень, звание, Ф.И.О.)

АИ

«04»

06

2019 г.

(подпись)

по безопасности жизнедеятельности:

д.х.н., проф., Приходько Н.Г.

(ученая степень, звание, Ф.И.О.)

НГ

«06»

06

2019 г.

(подпись)

Нормоконтролер: Чигамбаев Т.О.

(ученая степень, звание, Ф.И.О.)

ТО

«06»

06

2019 г.

(подпись)

Рецензент: к.ф.и.н., доц. Ракешева Б.Б.

(ученая степень, звание, Ф.И.О.)

ББ

«6»

06

2019 г.

(подпись)

Алматы 2019

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ»

Институт Космической инженерии и телекоммуникаций

Кафедра «Электроника и Робототехника»

Специальность 5В071600 - Приборостроение

ЗАДАНИЕ

на выполнение дипломного проекта

Студенту Белову Максиму Михайловичу

(Ф.И.О.)

Тема проекта Создание модуля распознавания лиц с Raspberry Pi

Утверждена приказом по университету № 124 от «26» октября 2018 г.

Срок сдачи законченного проекта «10» июня 2019 г.

Исходные данные к проекту (требуемые параметры результатов исследования (проектирования) и исходные данные объекта):

Создать модуль распознавания лиц, обладающий соответствующей базой данных лиц, а при ее отсутствии создает ее и вносит в нее новые лица, на видео-потоке выделяется область, где обнаружено лицо, а также процент вероятности совпадения с лицом из Базы.

Перечень вопросов, подлежащих разработке в дипломном проекте, или краткое содержание дипломного проекта: изучить искусственные нейронные сети и алгоритмы распознавания образов; разработать собственный алгоритм, идентифицирующий лица согласно базе данных лиц; произвести выбор подходящего программного обеспечения и библиотек; провести экспериментальное исследование с демонстрацией работоспособности системы.

Перечень графического материала (с точным указанием обязательных чертежей): Пример примитивов Хаара, метод обратного распростр. ошибки для обучения ИНС, структурная схема программы, схема подключения камеры и монитора к Raspberry Pi, архитектура сверточной ИС, содержимое папки с проектом, процесс работы программы, вектор имён, ошибочное определение, несрабатывание системы, корректная идентификация, корректное распознавание, исправление ошибки, команды для установки библиотеки, баланс рабочего времени на 2019 год, схема организации рабочего пространства, план помещения для расчётов по охране труда

Основная рекомендуемая литература: 1. Van Veen F. The neural network zoo The Asimov Institute, 2016. 2. Г.А. Кухарев, Н.А. Щеголева, Системы распознавания человека по изображению лица «ЛЭТИ», 2006. 3. Чабан А.И. Теория и алгоритмы распознавания образов. Учебное пособие. 4. Бекишева А.И. Метод. ук. к выполнению экон. части дипломной работы. 5. Налоговый Кодекс РК 25 декабря 2017г. №102-VI ЗРК, 6. Приходько Н.Г. Основы безопасности жизнедеятельности. Конспект лекций для студ. всех специальностей. 7. Документация Библиотеки OpenCV. 8. Самоучитель Python.

Консультации по проекту с указанием относящихся к ним разделов проекта

Раздел	Консультант	Сроки	Подпись
<u>Б.У.С.</u>	<u>Приходько Н.Г.</u>	<u>13.02 - 06.06.19</u>	<u>[Подпись]</u>
<u>Эконом. часть</u>	<u>Бекишева А.И.</u>	<u>16.5 - 4.06.</u>	<u>[Подпись]</u>
1. Теоретическое исследов.	Шабельников Е.А.	3.02 - 05.03	Шаб
2. Алгоритмы распознавания	Шабельников Е.А.	10.02 - 14.03	Шаб
3. Разработка ПО.	Шабельников Е.А.	02.03 - 14.04	Шаб

подготовки дипломного проекта


[illegible]

Дата выдачи задания « 1 » февраля 2019 г.

Заведующий кафедрой

_____ (Подпись) _____ (Чигамбаев Т.О.)
_____ (Ф.И.О.)

Научный руководитель
проекта

 (подпись) (Шабельников Е.А.) (Ф.И.О.)

Задание принял к
исполнению студент

Белов (подпись) (Белов М.М. (Ф.И.О.))

Аннотация

В ходе данной дипломной работы проведено исследование методов распознавания лиц на входящем потоке изображений, поступающих с камеры, то есть в режиме реального времени. В качестве основного метода для распознавания лиц был принят метод Виолы-Джонса и проведено теоретическое обоснование данного метода. Был выполнен подбор программы и библиотеки, на основе которых был создан собственный алгоритм идентификации людей на изображении. Проведено практическое исследование системы на примере базы из 7 человек. В результате проверки работоспособности программы были выявлены и исправлены ошибки алгоритма. Программа была практически реализована на базе Raspberry Pi. Приведены расчеты экономической эффективности проекта и охраны труда. В заключении приведены основные результаты работы и выводы на основе полученных результатов.

Андатпа

Осы дипломдық камерадан түсетін суреттердің, яғни нақты уақыт аралығында, тұлғаларды тану әдістеріне зерттеу жүргізілді. Виола-Джонс әдісі тұлға танудың негізгі әдісі ретінде қабылданды және осы әдісті теориялық негіздеу жүзеге асырылды. Бағдарлама мен кітапхана іріктелді, оның негізінде сурет салатын адамдарды анықтауға арналған алгоритм құрылды. 7 адамның базасында мысалға келтірілген жүйені практикалық зерттеу жүргізілді. Бағдарламаны тестілеу нәтижесінде алгоритмнің қателері анықталды және түзетілді. Бағдарлама Raspberry Pi негізінде жүзеге асырылды. Жобаның экономикалық тиімділігі мен еңбекті қорғау есептері келтірілген. Қорытындыда жұмыстың негізгі нәтижелері және алынған нәтижелерге негізделген қорытындылар келтірілген.

Annotation

In the course of this thesis, a study was conducted on the methods of face recognition on the incoming stream of images coming from the camera, that is, in real time. The Viola-Jones method was adopted as the main method for face recognition and the theoretical justification of this method was carried out. The program and the library were selected, on the basis of which he created his own algorithm for identifying people in the image. Conducted a practical study of the system on the example of a base of 7 people. As a result of testing the program, the errors of the algorithm were identified and corrected. The program was implemented on the basis of Raspberry Pi. The calculations of the economic efficiency of the project and labor protection are given. In conclusion, the main results of the work and conclusions based on the results obtained are presented.

Содержание

Введение	8
1 Теоретическое исследование.....	10
1.1 Искусственные нейронные сети(ИНС).....	10
1.1.1 Биологический прообраз ИНС	10
1.1.2 Нейрон.....	11
1.1.3 Построение нейронных сетей.....	14
1.2 Обучение нейронной сети.....	15
1.2.1 Концепция обучения.....	15
1.2.2 Метод обратного распространения ошибки.....	17
1.2.3 Гиперпараметры сети.....	21
1.3 Типы нейронных сетей для работы с изображениями	21
1.4 Свёрточные нейронные сети.....	22
1.4.1 Особенности и специализация.....	22
1.4.2 Принцип получения входного изображения.....	22
1.4.3 Принцип свёртки.....	31
2 Алгоритмы распознавания образов.....	33
2.1 Формальная постановка задачи.....	33
2.2 Этапы распознавания образов	33
2.2.1 Выделение образов.....	34
2.2.2 Логическая обработка.....	35
2.2.3 Принятие решений.....	36
2.3 Метод Виолы-Джонса.....	37
2.3.1 Особенности алгоритма.....	37
2.3.2 Алгоритм обработки.....	40
2.3.3 Усовершенствование алгоритма	41
3 Разработка программного обеспечения и его настройка.....	42
3.1 Установка среды разработки.....	42
3.2 Установка библиотек.....	43
3.3 Создание программы.....	44
3.3.1 Создание алгоритма и кода сохранения базы данных.....	44
3.3.2 Создание программы идентифицирующей лица.....	46
3.3.3 Проверка работоспособности.....	49
3.4 Запуск на Raspberry Pi.....	51
4 Экономическая эффективность.....	53
4.1 Расчёт трудоёмкости разработки программного продукта.....	53
4.2 Заработная плата.....	54
4.3 Материальные расходы.....	56
4.4 Расчёт электроэнергии.....	58

4.5 Амортизационные расходы.....	59
4.6 Накладные расходы.....	59
4.7 Себестоимость продукта.....	60
4.8 Цена реализации программного продукта.....	60
4.9 Оценка социально-экономических результатов функционирования программного продукта.....	61
5 Безопасность жизнедеятельности.....	64
5.1 Теоретические основы безопасности жизнедеятельности.....	64
5.2 Вредные факторы на производстве.....	65
5.3 Расчёт освещения рабочего места.....	67
5.3.1 Таблицы со значениями.....	67
5.3.2 Расчёт естественного освещения.....	70
5.3.3 Расчёт искусственного освещения рабочего помещения.....	74
Заключение	77
Список литературы	78
Приложение А	81
Приложение Б	82

Введение

Распознавание различных вещей на изображениях в последнее время стало очень актуально. Поиск по картинке, поиск похожих изображений, определение объектов. Однако наиболее интересной сферой стало распознавание лиц, которое распространяется повсеместно, начиная от развлечений, закачивая системами безопасности. Примером к первому является приложение Snapchat, где используются маски для редактирования фото. В пример ко второму можно привести китайскую систему «Скайнет», которая анализирует камеры наблюдения, установленные по городу, и помогает найти преступника.[1]

В Республике Казахстан подобной системы ещё не существует. Например, система «Беркут» сохраняет две фотографии: одна делается с отсканированного документа, удостоверяющего личность, а вторая делается непосредственно с человека. При этом проверка на соответствие этих фото возлагается на работника пограничной службы, что в свою очередь подвержено человеческому фактору. В то время как представленная в данной дипломной работе программа могла бы улучшить данный процесс, исключив работу человека, что является более качественным.

В любом случае задача по идентификации состоит из нескольких этапов. Первый этап – детектирование. Он заключается в поиске области на изображении, где находится лицо человека. Вторым этапом является экстракция признаков, то есть преобразование входных данных из детектированной области в вектор исходных признаков. Далее следуют селекция признаков или редукция пространства признаков. Последним этапом выполняется сравнение полученных признаков и базой данных эталонных признаков и определение класса объекта, если таковой имеется. В качестве класса в данном случае выступает личность человека.[2]

Но обработку данных могут усложнять нежелательные факторы, такие как шум, плохое освещение, и другие. Следует также заметить, что эти факторы, могут быть изменчивыми, что приводит к сложности составления математической модели для вычислений. И несмотря на то, что нам известна структура лица, расположение основных элементов может изменяться, например эмоциями или поворотом лица, что в свою очередь приводит к разнообразным неопределённостям на этапе обработки.[3]

По этой причине идентификация нуждается в достаточно качественном алгоритме обработки данных.

Большой вклад внесло использование нейронных сетей и машинного обучения для обработки биометрических характеристик лица человека. Машинное обучение выполняет распознавание с помощью нематематических

методик, то есть умение обрабатывать данные по абстрактным правилам или построение новых правил.[4]

На основе всего вышесказанного можно сделать вывод об актуальности исследования методов и алгоритмов идентификации лиц.

Основной целью данной дипломной работы является разработка программного продукта для распознавания лиц на основе выбранных в ходе исследования методов. Для достижения поставленной цели необходимо выполнить ряд задач:

- а) исследование и анализ методов распознавания лиц;
- б) разработать и реализовать алгоритм обработки видеофрагмента;
- в) создать базу данных лиц;
- г) провести практическое исследование работоспособности алгоритма;
- д) физическая реализация алгоритма на основе Raspberry Pi.

Настоящая работа включает в себя введение, пять глав, заключение, список литературы и два приложения.

Первая глава рассматривает теоретические основы нейронных сетей, а также рассматриваются методы получения цифровых данных фото-аппаратуры.

Вторая глава рассматривает основные алгоритмы распознавания образов на изображении. Отдельное внимание уделяется методу Виолы-Джонса.

В третьей главе произведена разработка программного обеспечения, направленного на распознавание и идентификацию лиц. В результате проведено экспериментальное подтверждение успешной работоспособности системы.

Четвёртая глава рассматривает экономическую эффективность проекта.

В пятой главе произведён анализ основных опасностей и вредных влияний на рабочем месте, рассчитано освещение на рабочем месте.

В заключении приведены результаты выполненной работы.

Приложения включают в себя коды программ по созданию базы данных и распознаванию лиц.

1 Теоретическое исследование

В первом разделе представлены теоретические основы принципов работы и методов обучения искусственных нейронных сетей, рассмотрены существующие типы нейронных сетей и приведено более подробное описание свёрточных нейронных сетей, которые используются в решении поставленных в работе задач.

1.1 Искусственные нейронные сети (ИНС)

1.1.1 Биологический прообраз ИНС

Первая модель нейронной сети была создана в 1943 году двумя учёными Уолтером Питтсом (Walter Pitts) и Уорреном Мак-Каллоком (Warren McCulloch), которые исследовали работу человеческого мозга, способы передачи данных и сигналов внутри мозга посредством нейронов. На основе этих исследований была построена первая простая машина, вычисляющая результат логических и арифметических действий.[5]

В машине имелись переключатели, имитировавшие работу биологических нейронов – клеток, которые получают, обрабатывают и передают информацию через электрические сигналы. На рисунке 1.1 представлена структура нейрона.

Типичная структура нейрона

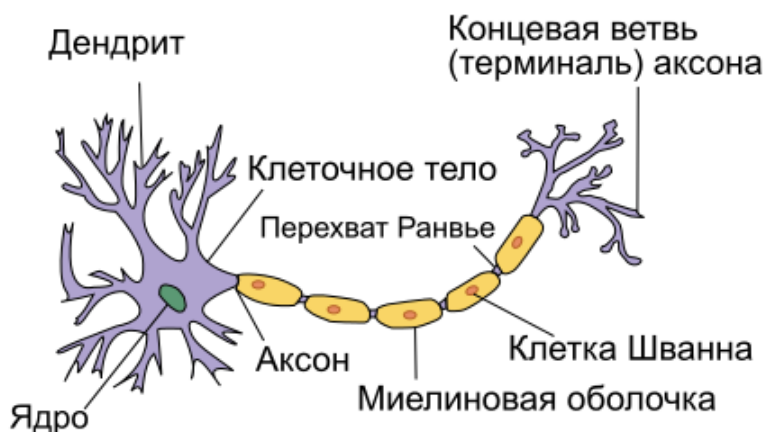


Рисунок 1.1 – Структура биологического нейрона

Так как искусственный нейрон – это математическая модель биологического нейрона, то он имеет схожие с ним функции:

- а) Приём информации из нескольких входов;
- б) Суммирование всех сигналов и их обработка;
- в) Передача результата на выход.

Представить искусственный нейрон можно в виде модели, изображенной на рисунке 1.2.

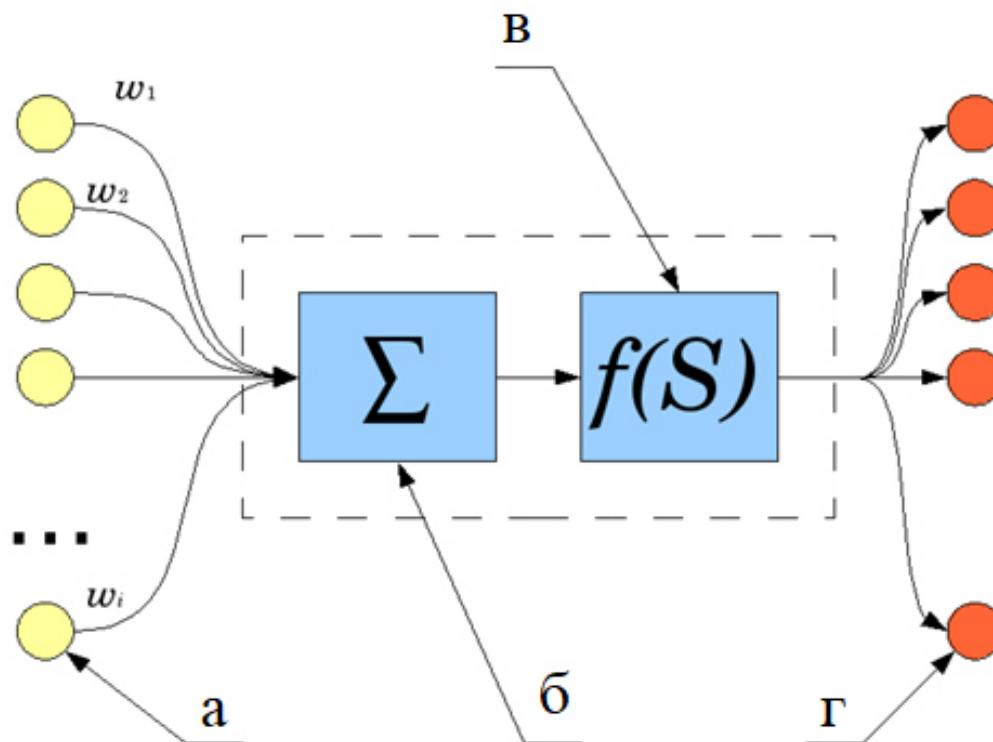


Рисунок 1.2- Схема искусственного нейрона

где: а – входные нейроны, сигналы которых идут на вход сумматора;
б - сумматор входных сигналов;
в - вычислитель передаточной функции;
г - нейроны, на входы которых подаётся выходной сигнал данного;
 ω – веса входных сигналов.

1.1.2 Нейрон

Нейрон начинает обработку полученных сигналов сразу после их поступления на вход.[6]

Математическая модель нейрона обладает нелинейной функцией активации, представленной в формуле(1.1), в которой аргументом выступает сумма всех входных сигналов:

$$F(X) = a_1x_1 + a_2x_2 + \dots + a_nx_n, \quad (1.1)$$

где: X – входные значения;
 a_n - весовые коэффициенты.

Функция активации характеризует зависимость выходного сигнала от суммы входных сигналов. Наиболее вероятно, что значения функции находятся в пределах $[-1,1]$, если это гиперболический тангенс, или $[0,1]$ для сигмоида. Важным является монотонное возрастание функции ввиду того, что некоторые алгоритмы, используемые в машинном обучении требуют непрерывно дифференцируемую функцию для всей числовой оси.

В 1957 году Френком Розенблатом (Frank Rosenblatt) был разработан перцептрон, модель искусственного нейрона, выход которого принимает только значения 0 или 1. Однако основным отличием от предыдущих моделей было наличие так называемых весов – численных значений, характеризующих важность каждого из входов и соответствующих им значений. Кроме того, взвешенная сумма всех входов сравнивалась с пороговым значением (threshold), от результата сравнения зависело значение на выходе. Система уравнений, описывающая модель перцептрона представлена в формуле (1.2) :

$$P(X) = \begin{cases} 0, & \sum_i w_i x_i < threshold \\ 1, & \sum_i w_i x_i \geq threshold \end{cases} \quad (1.2)$$

На рисунке 1.3 представлен соответствующий график.

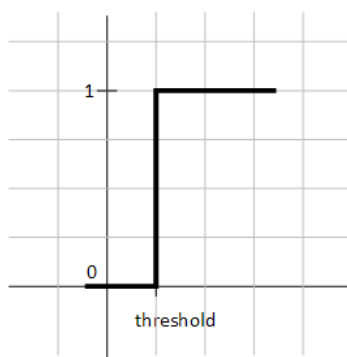


Рисунок 1.3 – График функции активации перцептрона

Но нейронные сети, основанные на перцептронах, сложно обучать. Ввиду того, что выход бинарный, то даже небольшое изменение одного из входов, если сеть достаточно большая, может привести к резким и непредсказуемым изменениям на выходе. Подобную сеть сложно анализировать. Однако, перцептроны являются базой для нейронов других типов, функция активации которых имеет более плавный график изменения состояния.[4]

По типу функции активации выделяют следующие типы нейронов:

а) Единичный скачок (частный случай перцептрона), формула (1.3):

$$P(X) = \begin{cases} -x, & x < 0 \\ x, & x \geq 0 \end{cases}, \quad (1.3)$$

б) Сигмоид, формула (1.4):

$$P(X) = \frac{1}{1 + e^{-aX}}; \quad (1.4)$$

в) Гиперболический тангенс, формула(1.5):

г)

$$P(X) = \tanh\left(\frac{x}{a}\right). \quad (1.5)$$

В каждой формуле a – угол наклона графика функции.

Сигмоид представляет собой развитый перцептрон. Выход перцептрона изменяется не скачком, а постепенно, с углом наклона a . График сигмоида представлен на рисунке 1.4.

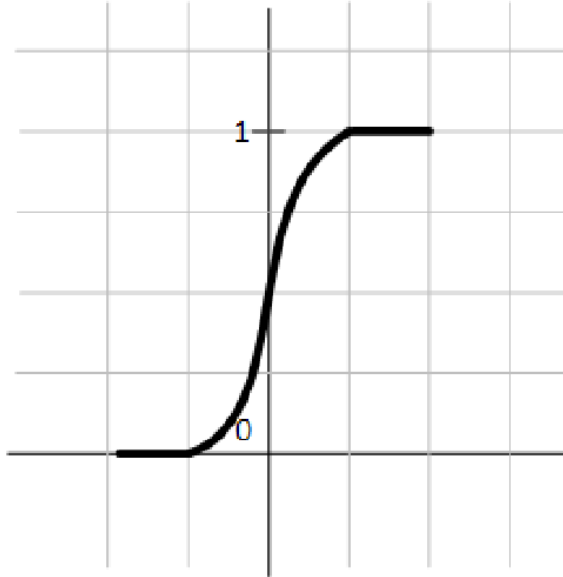


Рисунок 1.4 – График функции сигмоида

Однако наиболее распространён нейрон, график которого гиперболический тангенс, так как он захватывает отрицательные значения в диапазоне $[-1; 1]$, что наиболее удобно для обработки.

1.1.3 Построение нейронных сетей

Самостоятельно нейрон выполняет только одну функцию. Но для решения сложных логических, математических или других действий необходимо объединить нейроны в сеть.

Выделяют три типа нейронов, которые имеют одинаковую структуру, но разное назначение:

а) входные – необходимы для введения сигнала в сеть, имеют единственный вход;

б) выходные – могут иметь неограниченное конечное число входов и один выход, расположенный на периферии сети и выводящий результат вычислений на выход;

в) скрытые – находятся во внутренних слоях нейронной сети. Они принимают сигнал, обрабатывают его и передают дальше по сети;

г) нейроны смещения – позволяют сместить график функции по оси X , единственные нейроны, направленные назад, в предыдущий слой, и не имеющие весового коэффициента.

На рисунке 1.5 представлена структура простейшей нейронной сети. На рисунке обозначены I_k – входные нейроны, H_i – скрытые, и выходной нейрон - O .

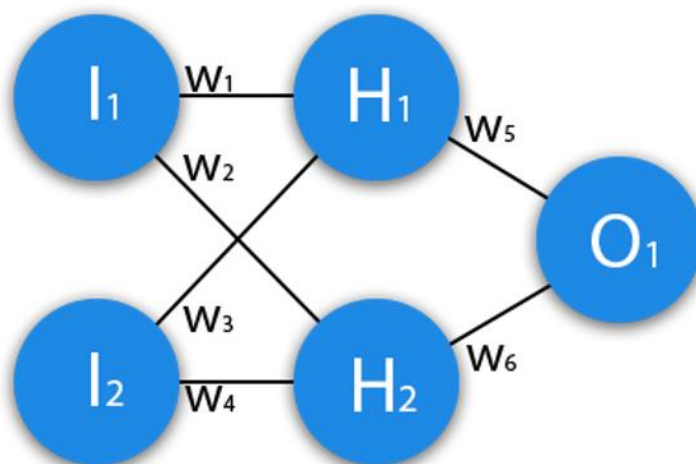


Рисунок 1.5 – Структура нейронной сети

На рисунке 1.5 имеются 2 входных нейрона, сигналы с которых поступают в скрытый слой. Данная сеть является сетью прямого распространения, так как сигнал проходит от начала до конца. Подобные сети используются для задач классификации. Следует заметить, что нейроны смещения допустимо не отмечать в схеме, но их нужно учитывать при вычислениях.

Таким образом для некоторой сети с n слоями, с учётом входных и выходных нейронов, с весами $W^{(1)}$ и функцией активации $A^{(m)}$ для каждого из слоёв, и нейронами смещения $b^{(1)}$ итерационная формула будет иметь вид формулы (1.6):

$$\begin{cases} z^{k+1} = W^{(k)} a^{(k)} + b^{(k)} \\ a^{k+1} = f(z^{(k+1)}) \end{cases}, \quad (1.6)$$

где: f - активационная функция;
 (W, b) – параметры сети.

1.2 Обучение нейронной сети

Различают три основных метода обучения: с учителем, без учителя и обучение с подкреплением. Обучение с учителем используется наиболее часто и для него необходимы тренировочные данные, для которых уже известны достоверные результаты. В качестве учителя зачастую выступает программа, с обучающей выборкой. Во втором типе программа-учитель отсутствует. Таким образом сеть самостоятельно разрабатывает алгоритм определения и классификации данных. При этом даже сами параметры определяются самим алгоритмом. Суть третьего типа заключается в том, что сеть самостоятельно ищет путь решения задачи, однако верные результаты закрепляются учителем.

После обучения сети всегда используется выборка данных для проверки работоспособности сети.

1.2.1 Концепция обучения

Сразу после создания сеть необходимо обучить на решение поставленной задачи. Обучение производится так называемыми тренировочными сетями, то есть наборами данных, результаты которых заведомо известны. При обучении вводится счётчик итераций, который растёт на единицу после каждого прохождения тренировочного сета. После нескольких итераций, количество которых определяется тренировочным множеством, но может зависеть и от иных параметров, оканчивается эпоха. Эпох может быть несколько, то есть обучение сети происходит многократно на неизменном обучающем множестве. [7]

Ввиду того, что результатом нейрона является линейная комбинация входных значений, то суть обучения сети сводится к математической задаче линейной оптимизации функции со множеством параметров. Параметр оценки многопараметрической оптимизации – это a , ошибка сети при достоверно верном ответе y . Задача оптимизации будет иметь следующий вид формулы (1.7):

$$r(w) = \sum_i C(a(x_i), y_i) \rightarrow \min, \quad (1.7)$$

где: C – функция потерь, определяющая ошибку при достоверном результате.

В качестве основного метода обучения ИНС используется метод градиентного спуска, суть которого заключается во внесении соответствующей поправки в величину веса сразу после каждого шага на значение формулы (1.8):

$$w^{(k+1)} = w^{(k)} - \eta \left(\frac{\delta r}{\delta w} \right). \quad (1.8)$$

Если функция риска $r(w)$ и активационная функция f дифференцируются, то при их подстановке получится формула (1.9):

$$w^{(k+1)} = w^{(k)} - \eta \sum_i C'_a(a(x_i), y_i) f'(\langle w, x_i \rangle) x_i. \quad (1.9)$$

При этом изменения значений весов происходят после прохождения всех итераций. При необходимости изменения значений весов после окончания каждого отдельного тренировочного сета, используется стохастический градиентный спуск, представленный в формуле (1.10):

$$w^{(k+1)} = w^{(k)} - \eta C'_a(a(x_i), y_i) f'(\langle w, x_i \rangle) x_i. \quad (1.10)$$

Каждый из представленных методов обладает своими плюсами и минусами. Первый метод обрабатывает весь объем данных одновременно, что в свою очередь требует большого объема оперативной памяти во время обработки, но при малом количестве данных является самым быстрым, ввиду матричного вычисления значений. Более простым методом является стохастический метод, он удобен при большом объеме данных и имеет возможность добавления данных непосредственно в период обучения, что делает обучение более гибким.

Однако на практике чаще используется метод мини-пакетов, комбинирующий преимущества обоих из вышеописанных методов. Этот метод заключается в том, что на каждой итерации из всех обучающих данных образуются случайные небольшие мини-пакеты, расчет которых производится по методу градиентного спуска.

Согласно требованиям стохастического и пакетного методов к функции потерь выдвигается требование, что она должна иметь вид формулы (1.11):

$$C = \frac{1}{n} \sum C_x, \quad (1.11)$$

где: C – общая функция потерь;

C_x – функция потерь для каждого тренировочного набора.

Функция потерь подбирается под каждую нейронную сеть индивидуально и эвристически, согласно решаемым задачам. Но чаще всего используются следующие две.

Квадратичная функция с параметрами – эта функция распространена чаще, ввиду простоты её вычислений. В представленной ниже формуле (1.12) $a(x_i)$ – реальное полученное значение выхода, y_i – изначальное известное верное значение для i -ой итерации:

$$C(x, y) = \frac{1}{2} \sum_i \|a(x_i) - y_i\|^2. \quad (1.12)$$

Но есть и более сложная функция, которая называется функцией перекрёстной энтропии и имеет формулу (1.13):

$$C(x, y) = - \sum_i (y_i \log(a(x_i)) + (1 - y_i) \log(1 - a(x_i))). \quad (1.13)$$

1.2.2 Метод обратного распространения ошибки

Прежде чем применить метод градиентного спуска к многослойным сетям, необходимо создать метод расчёта функции потерь относительно всей сети, а также дальнейшей корректировки весовых значений сети согласно с задачей минимизации. Для решения данных задач применяют метод обратного распространения ошибки.

Пусть имеется m тренировочных наборов. $a(x_i)$ – функция активации для элемента x_i и имеет зависимость от параметров W и b . Тогда функция потерь для m примеров будет иметь вид:

$$C(W, b) = \left(\frac{1}{m} \sum_i C(W, b, x_i, y_i) \right) + \frac{\lambda}{2} \sum_{l=1}^{n_l-1} \sum_{i=1}^{S_l} \sum_{j=1}^{S_{l+1}} (W_{ij}^l)^2. \quad (1.14)$$

В представленной формуле первое слагаемое представляет собой сумму квадратов ошибок, а вторая часть – это член регуляции уменьшения весов, направленный на препятствие переобучению нейронной сети. Здесь параметр λ оценивает значимость частей выражения по отношению друг к другу. Эту же формулу (1.14) можно представить в виде:

$$C(W, b) = \left(\frac{1}{m} \sum_i \frac{1}{2} \|a(x_i) - y_i\|^2 \right) + \frac{\lambda}{2} \sum_{l=1}^{n_l-1} \sum_{i=1}^{S_l} \sum_{j=1}^{S_{l+1}} (W_{ij}^l)^2. \quad (1.15)$$

На основании вышесказанного можно представить последовательность алгоритма обратного распространения ошибки следующим набором последовательных шагов:

а) выполняется проход сигнала через сеть в прямом направлении, на основании которого производится расчёт функций активации слоёв, вплоть до выходного слоя L_n ;

б) выполняется расчёт ошибки для каждого i -ого нейрона на выходном слое L_n по формуле (1.16):

$$\delta_i^{L_n} = \frac{\partial \left(\frac{1}{2} \|a(x_i) - y_i\|^2 \right)}{\partial z_i^{L_n}} = -(y_i - a_i^{L_n}) f'(z_i^{L_n}); \quad (1.16)$$

в) для каждого слоя в обратном направлении начиная от L_n и заканчивая слоем L_2 вычисляется ошибка:

$$\delta_i^{L_k} = \left(\sum_{j=1}^{S_{L+1}} W_{ji}^{L_k} \delta_j^{L_{k+1}} \right) f'(z_i^{L_n}); \quad (1.17)$$

г) производится расчёт производных для метода градиентного спуска:

$$\frac{\partial C_{wb}(x, y)}{\partial W_{ji}^{L_k}} = a_j^{L_k} \delta_j^{L_{k+1}}, \quad (1.18)$$

$$\frac{\partial C_{wb}(x, y)}{\partial b_i^{L_k}} = \delta_i^{L_{k+1}}. \quad (1.19)$$

Для повышения скорости работы при использовании матричных методов необходимо представление алгоритма в матричном виде, как последовательность шагов:

а) выполняется проход сигнала через сеть в прямом направлении, на основании которого производится расчёт функций активации слоёв, вплоть до выходного слоя L_n ;

б) рассчитывается матрица ошибок слоя L_n :

$$\delta^{L_n} = -(y - a^{L_n}) \circ f'(z_i^{L_n}), \quad (1.20)$$

где: \circ — покомпонентное произведение матриц;

в) для каждого слоя в обратном направлении начиная от L_n и заканчивая слоем L_2 вычисляется ошибка:

$$\delta^{Lk} = (W^{Lk} \delta^{Lk+1}) \circ f'(z_i^{L_n}); \quad (1.21)$$

г) производится расчёт производных для метода градиентного спуска:

$$\nabla_W C_{Wb}(x, y) = (a^{Lk})^T \delta^{Lk+1}, \quad (1.22)$$

$$\nabla_b C_{Wb}(x, y) = \delta^{Lk+1}. \quad (1.23)$$

В этих формулах $\nabla_W C_{Wb}(x, y)$ – градиент, или вектор, элементами которого являются частные производные $\frac{\partial C}{\partial W}$ или $\frac{\partial C}{\partial b}$ соответственно.

Теперь для обучения нейронной сети можно получить алгоритм путём комбинации представленного выше матричного алгоритма обратного распространения ошибки и мини-пакетного метода. В результате получится следующий алгоритм:

- а) инициализация базы, содержащей тренировочные сетки;
- б) для каждого входного объекта произвести прямое прохождение сигнала до конца, расчёт ошибки и применить обратное распространение ошибки;
- в) произвести балансировку новых значений весов по формуле (1.24):

$$w^{(k+1)} = w^{(k)} - \frac{\eta}{m} \sum_x \delta^{x,k} (a^{x,k-1})^T, \quad (1.24)$$

а также рассчитать смещение по формуле (1.25):

$$b^{(k+1)} = b^{(k)} - \frac{\eta}{m} \sum_x \delta^{x,l}. \quad (1.25)$$

Для оценки, верно ли настроена сеть и справляется ли с поставленной задачей, используют такой параметр, как сходимость – непрерывное убывание функции в процессе обучения. Используя метод обратного распространения ошибки, можно научить сеть с любого изначального состояния, что доказывает теорема сходимости перцептрона, которую описал и доказал Френк Розенблатт (Frank Rosenblatt). Согласно данной теории перцептрон, обучаемый данным методом, независимо от начальных значений весов за конечное время достигнет состояния, отвечающего за корректное решение задачи.[8]

На рисунке 1.6 – представлены 3 графика. График а) сходится, в то время как графики б) и в) не сходятся.

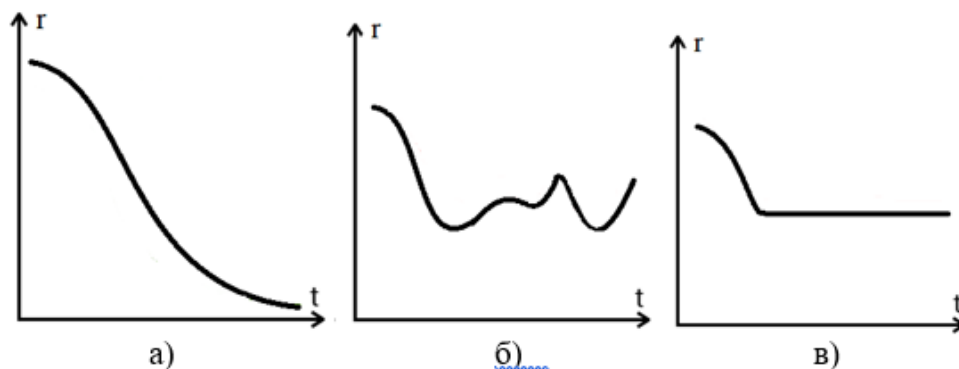


Рисунок 1.6 – Сходящийся и не сходящиеся графики

Однако следует отметить, что кроме достоинств у метода распространения обратной ошибки есть и недостатки:

а) некоторые активационные функции, которые имеют горизонтальные асимптоты, могут привести к зависанию сети. Если в момент коррекции значения веса представляют собой большие величины, то производная от их сжимающей функции стремится к нулю, что ведёт к уменьшению величины шага градиентного спуска и веса соответствующего нейрона. Нейрон не может выйти из состояния «паралича» самостоятельно. Но этого состояния можно избежать, уменьшая величину шага η ;

б) при наличии в сети большого количества весов сеть может прийти в состояние переобучения;

в) при уменьшении шага η пропорционально увеличивается время обучения, что приводит к медленной сходимости ряда;

На основании вышесказанного необходимо осторожно выбирать η , чтобы сохранить баланс скорости обучения и одновременно недопущения ввода нейронов в состояние «паралича». Однако, данные значения не вычисляются, а чаще всего подбираются путём подбора оптимальных значений.

Чтобы сеть не переобучилась используется ранняя остановка, заключающаяся в проверке точности выполнения задачи после каждой из эпох.[4]

В тот момент, когда точность перестаёт расти, следует остановить обучение, то есть подобран оптимальный ряд весов, и дальнейшее обучение может привести к состоянию переобучения, что в свою очередь ведёт к ошибочным результатам.

Чтобы процесс обучения был быстрее, темп обучения задаётся переменным, то есть в случае ухудшения точности проверки темп обучения снижается в 5-10 раз, после чего обучение продолжается, и так повторяется до конца обучения.

1.2.3 Гиперпараметры нейронной сети

Для регулирования работы и обучения нейронной сети используется ряд параметров, которые были описаны ранее. Как правило подбор параметров происходит эвристически, согласно назначению сети, другим параметрам и техническому заданию. К таким параметрам относятся:

а) величина пакета. Большой размер пакета означает редкое обновление значений весов, что не позволяет вовремя добавить новые объекты в обучающий сет;

б) скрытые слои. Как правило количество слоёв выбирается от 1 до 5 путём подбора;

в) скорость обучения. При регулировании размера шага η в градиентном спуске подбирается такое значение, когда наблюдается максимальная скорость, при минимальных колебаниях целевой функции. Как правило сначала $\eta=0.01$, и от этого значения ведутся соответствующие изменения, то есть увеличение, пока не возникнут колебания, или уменьшение, при слишком большом шаге;

г) параметр регуляции λ , определяющий влияние каждого отдельного тестового набора на обучение всей сети. Чем больше данный параметр, тем сильнее влияние тестового объекта на сеть. Однако, следует учесть, что слишком большое значение приводит к слишком большому изменению сети после каждой итерации, в то время как малое значение параметра ведёт к увеличению количества эпох. Как правило сначала данный параметр равен 1 и увеличивается в 10 раз после каждого шага, пока не нарушается сходимость целевой функции.

1.3 Типы нейронных сетей для работы с изображениями

Из всего многообразия нейронных сетей, целью которых является классификация, можно отдельно выделить те, основной задачей которых является обработка изображений. Наиболее часто используемые из них:

а) свёрточные нейронные сети (convolutional neural networks, CNN) – направлены на задачи поиска и классификации образов на изображении;

б) развёртывающие нейронные сети (deconvolutional networks, DN) – применяются при генерации изображений по исходным данным или изображениям с общим классификатором;

в) глубокие свёрточные обратные графические сети (deep convolutional inverse graphics networks, DCIGN) – направлены на создание изображений, обладающих несколькими классификаторами, на базе тренировочной выборки графических данных для всех классификаторов в частности. Может работать как в прямом, так и обратном направлениях;

г) глубинная свёрточная нейронная сеть (deep convolutional neural networks, DCNN) – является свёрточной нейронной сетью, имеющей на выходе сеть прямого распространения. Основной задачей является поиск образа и его классификация.

На основе данных типов нейронных сетей для выполнения данной работы и достижения поставленной задачи наиболее подходящими являются свёрточные и глубинные свёрточные нейронные сети.

1.4 Свёрточные нейронные сети

1.4.1 Особенности и специализация

Далее будут рассмотрены структура, свойства и гиперпараметры свёрточных нейронных сетей.

Специализация свёрточных нейронных сетей указывает на основные требования к их архитектуре. В изображении входом является каждый пиксель, для каждого из которых соответственно необходим свой вес. Например, изображению размером 100x100 пикселей с тремя каналами цвета необходимо $100 \cdot 100 \cdot 3 = 30\,000$ весов на скрытом уровне. С увеличением размера изображения количество весов увеличивается в разы, что значит усложнение обучения. Обучение таких сетей является нецелесообразным и сложным.

1.4.2 Принцип получения входного изображения

Как было отмечено выше для изображения входным сигналом является множество пикселей, полученных с устройства, регистрирующего видео. Современные системы регистрации видео представляют собой цифровую видеокамеру, блок обработки и хранения информации. То есть, как показано на рисунке 1.7, это цепь из объектива, цифровой матрицы, усилителя, аналого-цифрового преобразователя, процессора и устройства хранения информации. Единственная особенность заключается лишь в том, что в видео существует функция циклической записи. Эта особенность заключается в том, что каждый последующий кадр начинает записываться поверх старого, автоматически стирая последний.

Главное отличие цифровой камеры заключается в том, что видео информация, собранная на светочувствительном элементе, преобразовывается из аналоговой формы в цифровую, и впоследствии система уже работает с цифровым сигналом, то есть с набором нулей и единиц. Основной задачей для сбора видеoinформации является преобразовать свет в электрический сигнал.

Также нужно сделать это так, чтобы затем было возможно воспроизвести эту информацию, восстановить изображение по набору цифровых сигналов.

Как видно из схемы первым элементом является объектив. Это комплекс линз и диафрагм, который позволяет пропорционально изменить размеры реального изображения, чтобы каждая его часть равномерно легла на матрицу

видеокамеры. Так же объектив частично определяет угол обзора камеры. На рисунке 1.8 представлен объектив камеры.

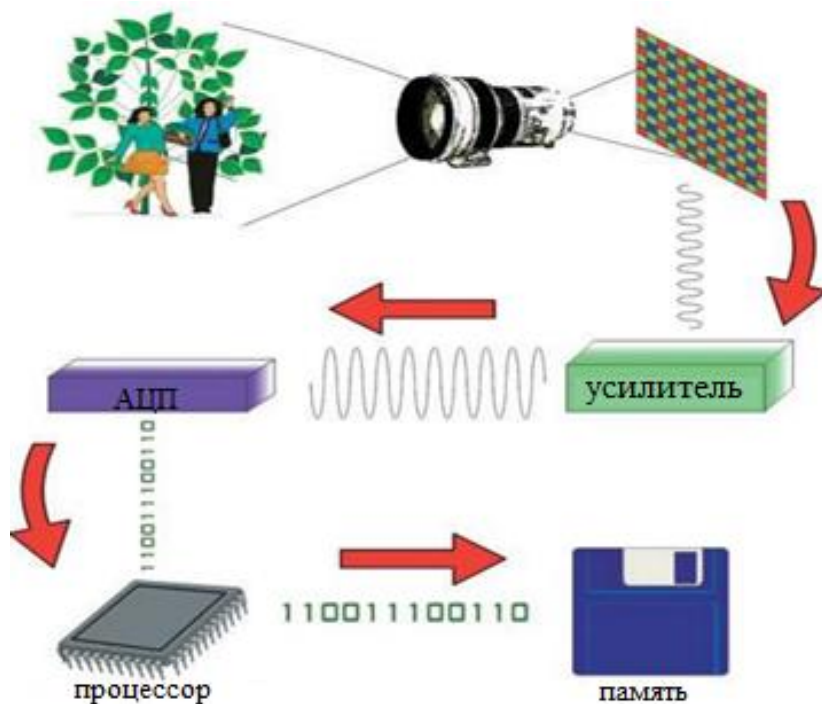


Рисунок 1.7 – Принцип работы цифровой записи изображений

Также нужно сделать это так, чтобы затем было возможно воспроизвести эту информацию, восстановить изображение по набору цифровых сигналов.

Как видно из схемы первым элементом является объектив. Это комплекс линз и диафрагм, который позволяет пропорционально изменить размеры реального изображения, чтобы каждая его часть равномерно легла на матрицу видеокамеры. Так же объектив частично определяет угол обзора камеры. На рисунке 1.8 представлен объектив камеры.



Рисунок 1.8 – Объектив камеры

Затем, потоки света, проходящие через объектив, падают на матрицу. При этом как правило на каждый пиксель падает разное количество света, следовательно, в каждом пикселе накапливается разное количество заряда, который в последствии усиливается, из аналоговой формы преобразуется в цифровую с помощью аналогово-цифрового преобразователя. После АЦП получается цифровой сигнал, который поступает в процессор, где он обрабатывается компрессируется и поступает в память для хранения и последующей обработки.

Один из способов произвести преобразование свет-сигнал основан на эффекте под названием фотоэффект. Суть этого способа заключается в том, что при падении светового потока на фотодиод или какой-нибудь другой фотоэлемент, возникает электрический ток, который также называют фототок. То есть, при спуске затвора, миллионы крошечных ячеек фоточувствительного элемента освещаются определенным количеством света, и преобразуют, этот свет в фототок независимо от цвета падающего света, так как большинство фотоэлементов не воспринимают цветность, а воспринимают только яркость. Также следует отметить, что сила тока будет зависеть только от интенсивности падающего светового потока. Следовательно, чем ярче свет, тем больше ток.

Для того, чтобы получить из данного светового потока цветное изображения используется процесс, который называется цветовой интерполяцией. Принцип действия данного процесса заключается в том, что программное обеспечение цифровой камеры анализирует полученную информацию с соседних ячеек, и по сложному алгоритму рассчитывает цветность каждой ячейки. Этот процесс является очень важным в получении итоговой картинке, так как он в равной степени с качеством светочувствительных элементов влияет на итоговое качество.

Исходя из того, что интенсивность падающего света и величина фототока связаны прямо пропорционально, между ними есть строгая математическая зависимость. И на основе данного преобразования основан принцип действия многих фоточувствительных элементов.

Однако существуют различные технологии изготовления светочувствительных сенсоров и в зависимости от технологии они по-разному осуществляют задачи накопления и хранения фотоэлектронов.

существуют две основные технологии преобразования свет-сигнал. Приборы с зарядовой связью, как правило, являются очень качественными. Это связано с тем, что они мало подвержены внутренним шумам, так как используют процесс зарядовой связи и передает только электрический заряд. Сам процесс зарядовой связи меньше подвержен шумам других электронных компонентов и обладает низким сопротивлением. Как следствие, выходной сигнал с ПЗС-датчика, обладает меньшим шумом по сравнению с КМОП датчиком. Однако недостаток этой технологии заключается в том, что при

передаче заряда между зарядовым колодцем и выходом датчика теряется часть электронов, и, следовательно, передача заряда не является на 100% эффективной.

Строение ПЗС-датчика представляет собой легированный полупроводник р-типа, тонкий слой полупроводника n-типа, тонкий слой диэлектрика, и электрод из поликристаллического кремния, который выполняет функцию затвора.

В качестве легированного полупроводника используется кремний р-типа. Легирование – это процесс добавления в чистый полупроводник атомов примеси, за счет которых повышается число дырок, и, следовательно, их становится в избытке. Соответственно, так как в полупроводнике р-типа основным носителем заряда являются дырки, то при подаче на затвор (электрод) положительного потенциала, все дырки начнут отталкиваться от него, под действием электрического поля, которое проникает в полупроводник через диэлектрический изолятор. Чем сильнее подать напряжения на затвор, тем сильнее будут отталкиваться дырки от него, и, следовательно, увеличится обедненная область (область где носители заряда отсутствуют). Как правило, основным носителем информации являются фотоэлектроны, следовательно, они накапливаются в образованной обедненной области, и несут информацию о яркости данного пикселя. Чем больше фотоэлектронов накопилось, тем выше яркость данного пикселя.

Наглядно принципиальное строение ПЗС-датчика показано на рисунке 1.9

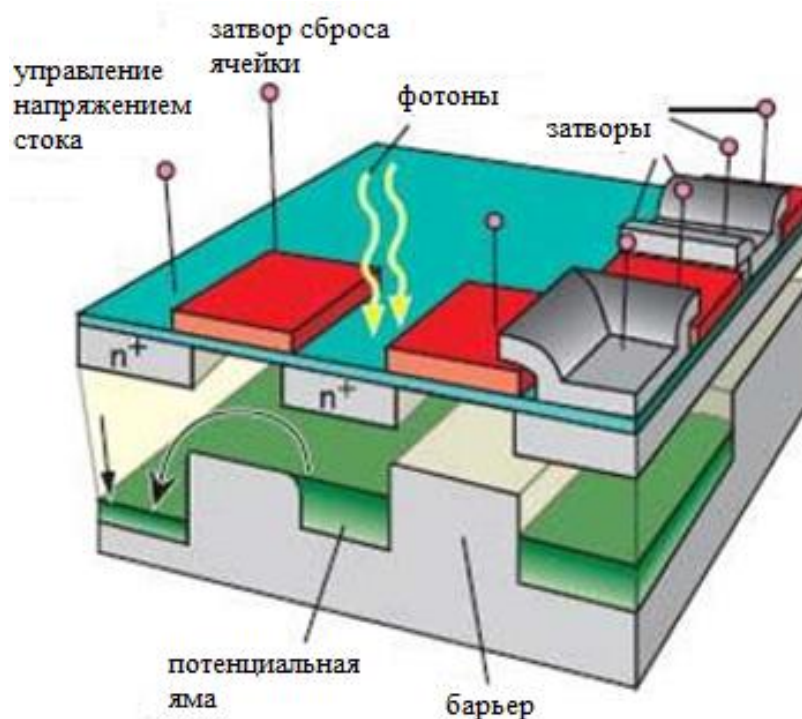


Рисунок 1.9 – Структура ПЗС-матрицы

Сам процесс протекания электронов происходит за счет изменения потенциала на соседних затворах. Например, если на соседнем затворе потенциал выше, то, следовательно, и потенциальная яма (обедненная область) там будет глубже, следовательно, фотоэлектроны (носители информации) перетекут туда, и так далее до усилителя.

Тонкий слой полупроводника n-типа служит для того, чтобы частично избавиться от влияния внутренних дефектов, следовательно, из-за его использования повышается четкость получаемого изображения. Также его использование снижает величину темного тока, который протекает по светочувствительному элементу, в отсутствие падающих фотонов.

Следующий тип датчика это КМОП-датчики. Следует отметить, что данный тип датчика построен по одноименной КМОП-технологии, а данная технология широко применяется для создания различных микросхем для огромного числа цифровых приборов. Отсюда можно сделать вывод, что одним из преимуществ данного датчика будет являться более низкая цена, чем на ПЗС датчики, для производства которых в свою очередь необходима дорогостоящая производственная база и специализированные технологические процессы.

Принципиальное отличие КМОП-датчика от ПЗС-датчика является процесс считывания информации. В КМОП матрицах можно считывать информацию с каждого пикселя отдельно и независимо. Из этого условия можно выделить еще несколько дополнительных преимуществ, помимо малой стоимости. Первая из них – это особенность строения датчика, которая позволяет расположить блок обработки информации прямо на кристалле. Также можно сказать и о том, что КМОП-датчики используют меньшее количество дополнительных компонентов, и как следствие они являются менее габаритными, чем ПЗС-датчики. Значительной особенностью является то, что КМОП-датчики не нуждаются в большом количестве энергии, следовательно, они являются более дешевыми в эксплуатационном плане.

Стоит отметить, что в качестве фотоэлемента используется подложку из кремния p-типа, как и в ПЗС-датчиках. Внутри образуется потенциальная яма, в которую в последствии будут попадать наши электроны. Количество этих электронов будет также зависеть от силы падающего света. А извлечение информации о падающем свете из пикселя в технологии КМОП сильно отличается от технологии, используемой в ПЗС-датчиках.

Как уже говорилось ранее, мы можем считывать информацию независимо из каждого пикселя. Это возможно за счет строения самого пикселя, а в частности за счет транзистора сброса. Технология считывания информации в КМОП-технологии заключается в том, что при срабатывании затвора на транзистор сброса поступает сигнал сброса и, следовательно, он открывается. Все накопившиеся фотоэлектроны протекают по нему на выход датчика. Таким образом мы можем считывать информацию с любого пикселя независимо и в

любом порядке, и информация из пикселя сразу поступает на выход датчика для последующей обработки.

Развитие данной технологии связано с тем, что мы можем расположить несколько транзисторов в одном пикселе. Как следствие стало возможным производить обработку полученной информации сразу внутри датчика.

Достаточно лишь интегрировать на данные микросхемы те устройства, которые нам требуются (например, усилитель).

Структура КМОП датчика представлена на рисунке 1.10.

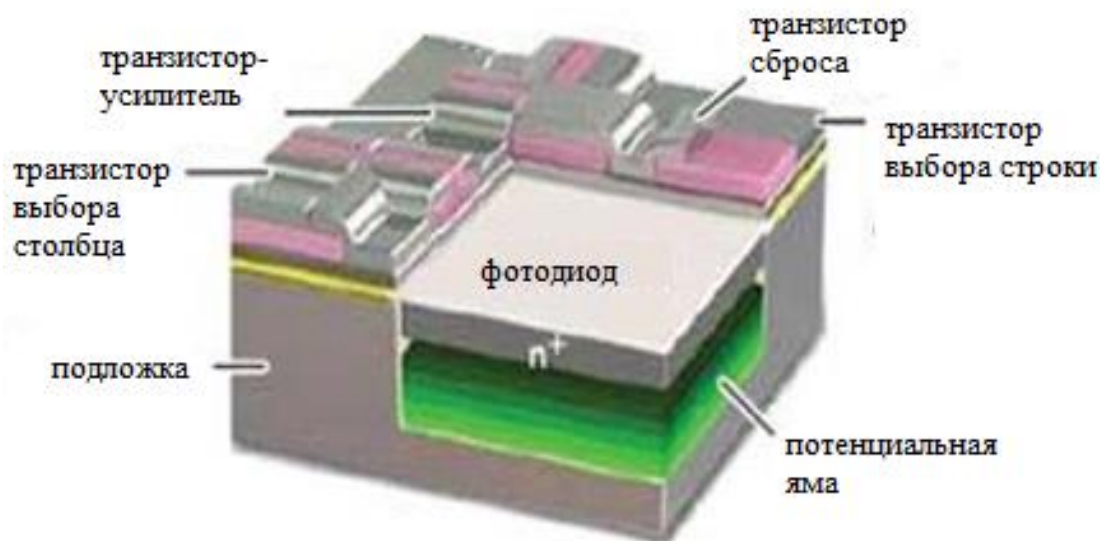


Рисунок 1.10 – Структура КМОП-матрицы

Как уже говорилось ранее, мы можем считывать информацию независимо из каждого пикселя. Это возможно за счет строения самого пикселя, а в частности за счет транзистора сброса. Технология считывания информации в КМОП-технологии заключается в том, что при срабатывании затвора на транзистор сброса поступает сигнал сброса и, следовательно, он открывается. Все накопившиеся фотоэлектроны протекают по нему на выход датчика. Таким образом мы можем считывать информацию с любого пикселя независимо и в любом порядке, и информация из пикселя сразу поступает на выход датчика для последующей обработки.

Развитие данной технологии связано с тем, что мы можем расположить несколько транзисторов в одном пикселе. Как следствие стало возможным производить обработку полученной информации сразу внутри датчика.

Достаточно лишь интегрировать на данные микросхемы те устройства, которые нам требуются (например, усилитель).

Следует отметить, что у КМОП-датчиков есть определенные недостатки. Из-за того, что все операции протекают на одном пикселе сразу, это приводит к

появлению дополнительных шумов при работе дополнительных схем, расположенных на кристалле. Для устранения этого недостатка перед светочувствительными элементами поставили собирающие свет микролинзы, а сами светочувствительные элементы стали располагать на обратной стороне от проводников и транзисторов. Также следует отметить, что современные КМОП-сенсоры нуждаются в кадровом затворе. При использовании строчного затвора изображение движущихся объектов будет размываться, так как он приводит к построчному смещению изображения и в каждый момент времени будет фиксироваться только часть этого изображения. Кадровый затвор позволяет избежать данного эффекта. Он экспонирует все пиксели сенсора одновременно, в течение равного промежутка времени. Реализация этого немного усложняет структуру нашего датчика. Так как мы экспонируем все пиксели одновременно, то нам следует добавить элемент памяти в каждый пиксель. Как правило, этим элементов памяти служит дополнительный конденсатор. Это ведет к тому что размер нашего пикселя увеличивается. Но КМОП-технологии развиты достаточно, чтобы сделать размер пикселя приемлемым для съемки и в Full HD-качестве.[9]

Как говорилось изначально, любой датчик может воссоздать только черно-белое изображение. То есть он лишь показывает яркость падающего на каждый пиксель света. Если свет очень яркий, то цвет будет стремиться к белому, а если свет не падает на пиксель вообще, то мы получим черный цвет. Для того чтобы наша цифровая камера могла построить цветное изображение придумали перед каждым пикселем ставить светофильтры, которые пропускают свет определенного цвета. Как правило используются светофильтры зеленого, синего и красного цветов. Также их расположение зависит от выбранной модели расположения светофильтров. Как правило одной из основных таких моделей является модель Байера, которую также называют GRGB, RGBG, или (если надо подчеркнуть диагональное расположение красного и синего пикселей) RGGB, которая показана на рисунках 1.10-1.11.

Как видно из рисунка в модели Байера светофильтры разных цветов расположены в шахматном порядке, при этом зеленых светофильтров в два раза больше чем синих или красных. Это обусловлено тем что человеческий глаз лучше реагирует на зеленый цвет. Принцип действия данного фильтра заключается в том, что через каждый фильтр проходит только одна треть всего светового потока, а остальная часть отсекается, и при этом пиксели будут воспринимать падающий свет как черно-белый. Соответственно эта часть светового потока, которая прошла после фильтрации, будет характеризовать количество света, того цвета, что и светофильтр, через который прошел наш свет. Затем берутся четыре соседние ячейки 2x2 (красная, зеленая, синяя, зеленая) и создаются четыре черно-белых канала. Информация из этих ячеек

поступает в процессор по данным каналам, и в результате интерполяции рассчитывается цвет каждой ячейки.[10]

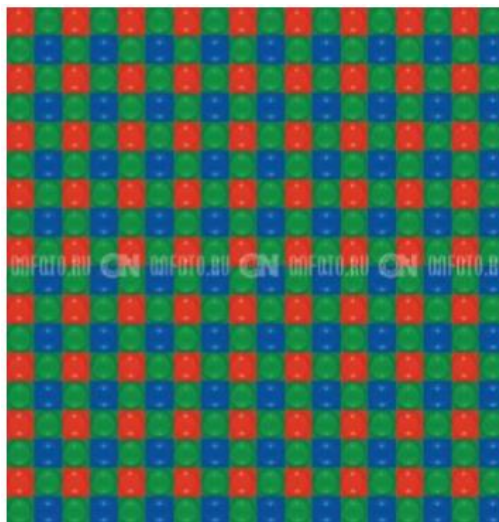
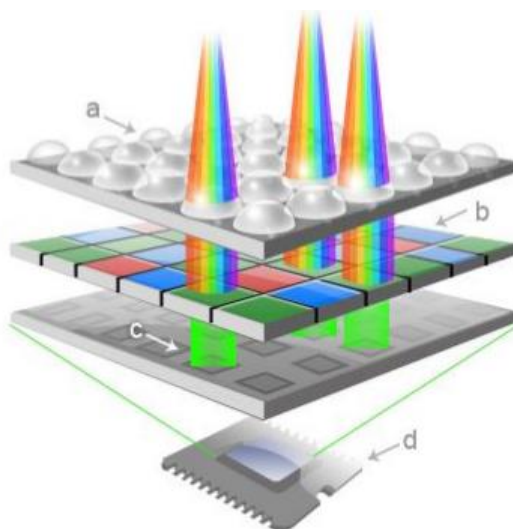


Рисунок 1.10 – Светофильтр RGBG



a – Микролинзы, b – Фильтр цвета, c – Фотодиод, d – Миллионы световых сенсоров

Рисунок 1.11 – Принцип работы RGBG светофильтра

Проблема любого фильтра «Байера» в том, что он склонен, снижать разрешение и качество изображения, в первую очередь, образуя искажения изображения. Это явление образуется при наложении двух структур с регулярным рисунком. По сути, это биение частот, а цветной рисунок возникает как раз именно из-за чередования цветных ячеек на фильтре. Чтобы уменьшить этот эффект, в 90% камер перед сенсором ставят специальный оптический

фильтр нижних пространственных частот (ОФНПЧ) "АА" (anti aliasing), размещаемый перед элементами матрицы, который размывает изображение. Естественно при этом сильно теряется разрешение изображений, получаемое путем интерполяции, но зато в какой-то степени уходит искажение.

Количество пикселей - это разрешение нашего изображения. Оно определяет количество точек, которое впоследствии мы увидим на экране. Однако это еще не говорит о том, что чем больше разрешение матрицы, тем обязательно качественней будет наша картинка. Может быть так, что мегапикселей на матрице много, а качество снимка все равно невысоко: он не резок, не контрастен, утопает в цифровом шуме — помехах. Качество изображения зависит не только от разрешения в мегапикселях, но и от физического размера самой матрицы. Именно за мелкие детали должно отвечать высокое разрешение матрицы.

Также немало важным является и размер этих пикселей. Он определяется двумя характеристиками: физический размер матрицы и количество пикселей. Чтобы получить размер матрицы нужно ее размер поделить на количество пикселей, и мы получим размер одного пикселя. Доказано, что размер пикселя (расстояния между пикселями) уже достиг своего физического предела, который составляет 1 мкм. Дальнейшее уменьшение размера пикселя теряет смысл, поскольку длина световой волны составляет от 0,39 до 0,78 мкм; и при расстоянии между пикселями менее 0,78 мкм (красный свет), соседние пиксели будут показывать одно и то же.

Чувствительность - под чувствительностью понимается отношение значения электрического сигнала, вырабатываемого датчиком, к уровню его освещённости в данный момент. Значение электрического сигнала измеряется в вольтах, а освещенность в люксах в секунду. Для цветных датчиков изображения значение чувствительности указывается отдельно для разных длин волн (монохроматическая чувствительность к потоку излучения шириной 1 нм). Также может использоваться суммарная чувствительность во всем спектральном диапазоне (болометрическая чувствительность). У многих черно-белых датчиков очень высока чувствительность в инфракрасном диапазоне. Спектральная чувствительность цветных датчиков изображения близка к чувствительности человеческого глаза.

При назначении той или иной чувствительности на самом деле в матрице никаких изменений не происходит, а изменяется коэффициент аналогового усиления сигнала с фотодатчиков перед подачей его на вход аналого-цифрового преобразователя. Чем больше выбираемая чувствительность отличается от реальной светочувствительности матрицы, тем сильнее сигнал. Но усиление сигнала может негативно сказаться на итоговом изображении — появляется «шум» в виде случайных помех.

Динамический диапазон - это отношение максимального выходного сигнала датчика к его собственному уровню шума, дБ. Человеческий глаз имеет динамический диапазон около 200 дБ. Ни один искусственный прибор не обладает столь высоким значением этого параметра.

Отношение сигнал/шум - это величина, которая показывает во сколько раз наш полезный сигнал превосходит шум. Для оптимальной картинке эта величина должна быть не меньше чем 45 дБ. При таком отношении сигнал/шум мы не будем замечать искажения на картинке, вызванные шумами.

Шум в матрице определяется случайным (тепловым) движением электронов, а сигнал интенсивностью светового потока, временем экспозиции (накопления заряда) и площадью светочувствительного элемента. Соответственно, чем выше параметры, образующие сигнал, тем будет лучше соотношение сигнал/шум при прочих равных условиях.

Шумы в КМОП-матрице неизбежны, так как каждый пиксель помимо светочувствительного элемента обладает своей схемой обработки сигнала. Соответственно, из-за того, что все пиксели расположены очень близко друг к другу, влияние цепей каждого пикселя неизбежны. Также стоит отметить, что ток, уходящий с пикселей также оказывает свое влияние. Устранение данной проблемы корректируются, в основном, внутренней разводкой КМОП-схемы. Когда ячейка матрицы сброшена в начальное состояние, ее емкостной элемент полностью заряжен, и этот заряд утекает разными путями. Существует несколько факторов, влияющих на такой ток.

Также следует сказать, что из каждой ячейки усиливается полученная видеоинформация и соответственно будет усиливаться и шум.

Шум может возникать и из-за естественного колебания температуры внешней среды. А конструкция матриц такова, что на каждые 10°C увеличения температуры происходит удвоение внутреннего шума.

1.4.3 Принцип свёртки

На рисунке 1.12 представлена упрощённая схема архитектуры свёрточной нейронной сети.

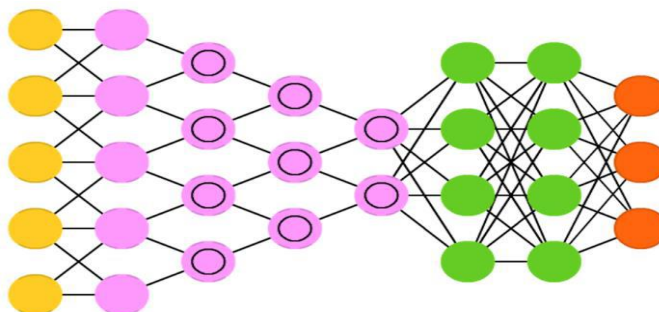


Рисунок 1.12 – Архитектура свёрточной НС

Слева находятся входные нейроны, представленные сигналами соответствующих пикселей. Далее происходит свёртка на сужающемся каскаде нейронов, расположенных на внутренних слоях. В правой части находится нейронная сеть прямого распространения и выходные нейроны.[4]

С каждым новым слоем количество нейронов уменьшается, для удобства вычислений уменьшение пропорционально степени числа 2. Следует отметить операция свёртки применяется уже на этапе считывания изображения: свёрточная матрица проходит по изображению из верхнего левого угла, сдвигаясь на один пиксель за шаг. При этом матрица весов для различных нейронов не изменяется, а результат прохождения матрицы по полю проходит этап линейной ректификации, целью которой является отсечение лишних данных во входном канале.

Слои свёртки идут поочерёдно со слоями субдискретизации, на которых происходит уменьшение размерности входящих карт признаков. Из нескольких рядом стоящих нейронов выбирается самое большое значение, которое становится новым нейроном на новой, создаваемой таким путём карте признаков.[7]

Пример свёртки представлен на рисунке 1.13.

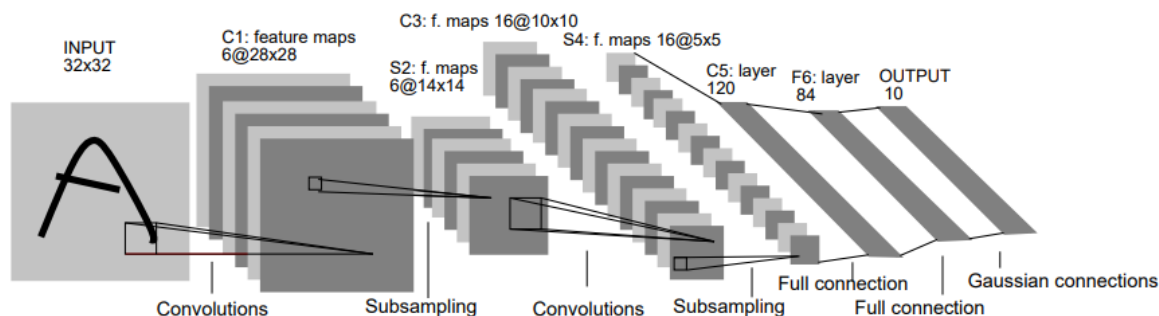


Рисунок 1.13 – Свёртка изображения

В результате каждого слоя свёртки и субдискретизации уменьшается размер изображения, что в конце даёт на выход сети малое число параметров, определяющие абстрактные понятия самой сети. Данные подаются далее на входы подключённой далее сети прямого распространения, которая выполняет обработку результата.

После подобной обработки входного изображения размером 100x100 с 3 каналами цвета, при ядре свёртки размером 3x3 и 6 каналами на выходе сети необходимо найти $9 \cdot 3 \cdot 6 = 162$ параметра.

2 Алгоритмы распознавания образов

Во втором разделе приводятся примеры различных алгоритмов и методов распознавания объектов на изображении, как математических, так и на основе нейронных сетей. Далее ставится формальная задача по распознаванию образов и приводится подробное рассмотрение метода Виолы-Джонса.

2.1 Формальная постановка задачи

Для постановки задачи предварительно нужно ввести понятие «образ». Образ – это объект, который характеризуется конечным набором признаков. Таким образом «распознавание образов» - это поиск, определение основных признаков объекта из общей информации о данном объекте, и определение принадлежности объекта к какому-либо из классов, то есть его классификация.[11]

Рассмотреть задачу по распознаванию образов можно с двух позиций. Первая позиция относится к теории ИНС, где используется метод получения конечного результата путём проведения эксперимента [12]. Вторая позиция принадлежит к области математики, где основу составляют рассуждения и доказательства, основанные на логике и математических методах.

Согласно вышесказанному постановку задачи можно сформулировать так: на входе имеется множество объектов и описание каждого из них. Имеются описания классов. Множество классов делит входное множество на подмножества. Необходимо выполнить классификацию входного множества объектов.

В качестве входного множества используется библиотека, которая содержит изображения, представленные в виде функций. Для облегчения расчётов входные изображения представляются в монохромном формате при условии, что цвет не несёт важной информации и не является одной из определяющих характеристик. В таком виде изображение представляется в виде множества точек на некоторой плоскости S , где $B(x,y)$ – это функция, определяющая яркость точки с координатами x и y . [13]

По сути задачу по распознаванию образов можно свести к оценке относительной вероятности принадлежности объекта к каждому из имеющихся классов. Решение находится путём исследования объекта на наличие у него уникальных свойств, присущих одному из классов. Классификация объекта происходит по определению наивысшей вероятности его принадлежности к тому или иному классу.

2.2 Этапы распознавания образа

Здесь рассмотрены алгоритмы подготовки входных данных, поиска образов на изображении, методы обработки полученного результата поиска, а также их плюсы и минусы.

2.2.1 Выделение образов

Для распознавания образов на исходном изображении необходимы идеальные условия, ввиду наличия различных шумов, которые приводят к погрешностям и ошибкам. Поэтому для использования реальных изображений предварительно необходимо произвести обработку изображения и произвести предварительный поиск области, где есть объекты.

Для поиска областей с объектом и отсеечения ненужных областей используются алгоритмы, называемые фильтрами. Чаще всего они применяют ко всем точкам единое преобразование. Ниже представлены фильтры, от самого простого к сложному:

а) бинаризация изображения по порогу. Данный способ является самым простым и быстрым, он использует значение определённого цвета на изображении, при этом на гистограмме отмечается некоторое пороговое значение этого цвета, и все цвета, которые не подходят, оказываются ниже порога, дальше просто не отображаются. Данный метод подходит для поиска на изображении предметов и зон одного цвета, выделения объектов на однотонном фоне;

б) фильтры низких и высоких частот. К обработке изображений можно применить преобразование Фурье и фильтры низких и высоких частот, используемые при обработке цифровых сигналов, так как изображение является сигналом сложного типа. После преобразования используется свёртка с готовым фильтром. В результате такой обработки изображения избавляются от шумов, или выносят их как отдельный канал информации;

в) вейвлет-преобразование очень похоже на предыдущий метод, но для свёртки применяются модели паттернов. Тогда основной задачей алгоритма выступает выбор правильных паттернов. К классическим паттернам для анализа данным методом являются: вейвлеты Хаара, Добеши, Мейера и мексиканская шляпа, которые представлены на рисунке 2.1;



Рисунок 2.1 - Вейвлеты Хаара, Добеши, Мейера, мексиканская шляпа

г) фильтрации функций – это математический метод, в основе которого лежит поиск функций, которые порождают каждую из множества точек исходного изображения. Для создания рисунка на основе математических функций используются окружности, параболы и прямые. Классическим для данного метода считается преобразование Хафа, или преобразование Радона, которое использует для интерполяции быстрое преобразование Фурье;

д) фильтрация контуров – это набор фильтров, который позволяет найти на изображении точки с максимальным отличием значения цвета от соседних. При соединении множества точек выделяет контуры любых участков объекта, при условии, что они отличаются от фона изображения. К таким фильтрам можно отнести оператор Кэнни, Собеля, Лапласа и др. Чаша применяются к изображениям с высоким разрешением.

Все вышеуказанные методы могут использоваться как самостоятельно, так и в комплексе с другими, подобное обусловлено техническим заданием.

Основным недостатком считается значительное увеличение времени соответственно увеличению размера изображения на входе.

2.2.2 Логическая обработка

После наложения на данные фильтров их можно использовать для последующей логической обработки. Формальная задача обработки формулируется так: переход от набора точек к объектам или их характеристикам. Далее приведены основные методы таких переходов:

а) математическая морфология – это непосредственное использование теории множеств, а также логических операций над множествами, к множеству пикселей входного изображения. Дополнением к исходному изображению становится некоторое количество специальных паттернов, которые являются бинарными, так как пиксели двухцветны. Дальнейшая работа над изображением происходит через логические операции над обоими множествами пикселей. К таким логическим операциям относятся: эрозия, замыкание, наращивание и размыкание. С помощью морфологической обработки находят границы объекта, убирают шум, удаляют лишние объекты малого размера и другие действия, поэтому чаще этот метод применим для работы в графических редакторах;

б) контурный анализ – метод, обладающий простой логикой, что многие объекты можно определить по их контурам, как, например, самолёт. Минусом данного метода является то, что для точного определения ему необходимы идеальные условия;

в) отслеживание специальных точек – к особым точкам относится множество объектов, которое полностью зависит от поставленной задачи и непосредственно объекта поиска. В качестве таких точек могут быть приняты локальные экстремумы на изображении, максимумы дисперсии, резкие углы и другие объекты и их параметры. Этот метод применим при отслеживании

движения в видеопотоке, объединения изображения с разных камер, стабилизация изображений. Но широкий диапазон использования приводит к сложности вычислений и необходимости большого множества точек, чтобы распознать объект более корректно. Таким множеством точек должны быть характеристики, точно идентифицирующие объект, то есть уникальные. Поэтому при сложности вычислений данный метод является наиболее точным, ввиду большого количества информации.

2.2.3 Принятие решений

На основе распознанных свойств представленного на изображении объекта производится его дальнейшая классификация. Для принятия верного решения применяются принципы обучения, которые были описаны в предыдущих главах.

Перед началом обучения имеющиеся данные логического анализа сопоставляются с исходными данными обучения.[6] Имеется некоторое пространство признаков, принадлежащее исходному изображению. Каждый из представленных признаков имеет собственный вес, то есть количественное представление его важности для принятия конечного решения. В процессе обучения всё пространство признаков делится на области, каждая из которых принадлежит к одному из решений. После каждой итерации происходит очередное деление и областей становится больше. Причем это дробление происходит не из-за изменения соотношения площадей долей каждого признака в общем пространстве, а ввиду правильного перераспределения областей пространства для каждого признака.

Ниже представлен пример выбора, результатом которого может быть либо 1, либо 0. Например, при нахождении объекта на изображении. На рисунке 2.2 представлено последовательное разделение пространства на области при различном количестве итераций, для 1, 7 и 40 итераций соответственно, а также представлен график сходимости ошибки.

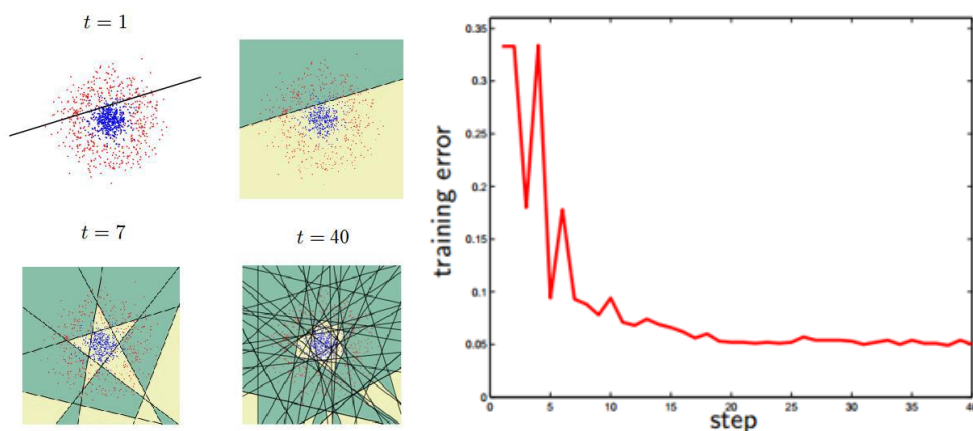


Рисунок 2.2 – Разделение пространства на области и график сходимости

2.3 Метод Виолы-Джонса

На сегодняшний день одним из наиболее эффективных алгоритмов распознавания является метод Виолы-Джонса, впервые реализованный в 2001 году Полом Виолой (Paul Viola) и Майклом Джонсом (Michael Jones)[14]. Его превосходство заключается в скорости распознавания, точности до 90% и малом количестве ложных срабатываний. Основным недостатком является распознавание лиц, повернутых максимум на 30 градусов. При произвольном угле наклона вероятность верного распознавания резко снижается[15]. Ещё одним недостатком можно указать долгое обучение сети и её классификаторов при использовании данного метода. [16]

Для реализации данного метода используются принципы, часть которых уже рассмотрена выше, а вторая часть будет описана далее:

- а) принцип сканирующего окна;
- б) интегральное представление исходных изображений;
- в) использование вейвлетов Хаара, для обнаружения объекта;
- г) применение бустера AdaBoost;
- д) использование бинарного классификатора;
- е) с использованием каскадов признаков окна, в которых лица не обнаружены, отсеиваются.

2.3.1 Особенности алгоритма

Принцип сканирующего окна был рассмотрен ранее. Входное изображение представляет собой матрицу с размером $m \times n$, где каждый элемент объединяет в себе значения характеристик цветового пространства RGB, либо может представляться в виде значения от 0 до 255^3 . В последнем случае представленное значение является результатом склеивания всех трёх компонент в одно число. В процессе обработки сканирующее окно передвигается по изображению. На каждой итерации обнаружение лица происходит путём запуска классификатора внутри сканирующего окна, и в случае обнаружения лица, окно с лицом выделяется. Сканирующее окно представляющее собой прямоугольник, имеющий изменяющийся размер, в начальный момент не превышающий размер самого изображения. Основными параметрами являются x, y – координаты центра окна, w, h – ширина и высота сканирующего окна, α – угол наклона.[16]

При этом входное изображение используется в интегральном виде, то есть в виде матрицы, имеющей тот же размер, но все элементы данной матрицы рассчитываются как суммарная интенсивность пикселей, находящихся левее и выше того, который необходимо рассчитать. В результате данного расчёта возможно рассчитать интенсивность любого случайного прямоугольника матрицы за ограниченное время. Для примера возьмём произвольный прямоугольник, представленный на рисунке 2.3.

Формула расчета его яркости:

$$L(ABCD) = L(A) + L(C) - L(B) - L(D).$$

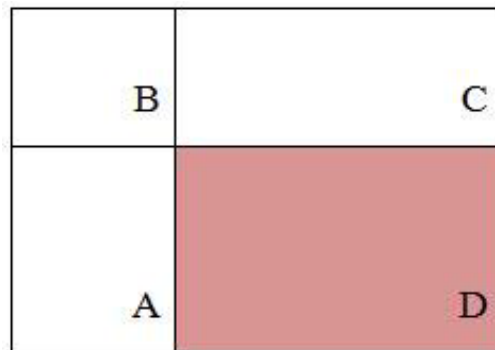


Рисунок 2.3 – Случайно выбранный прямоугольник в матрице

Как основные признаки классификации объектов, обнаруженных на изображении, применяются вейвлеты Хаара. Объект представляется как вектор признаков $x=(f_i(x))$, это признаковое описание объекта. Каждый из признаков имеет область допустимых значений D_f . Тогда признаковое пространство для сравнения с поисковым окном будет иметь вид множества:

$$X = D_{f1} * ... * D_{fn}.$$

В качестве признаков могут выступать примитивы, как в нынешнем алгоритме применяются расширенные примитивы Хаара, суть которых сводится в наложении примитива на выбранную область. Пример примитивов представлен на рисунке 2.4.

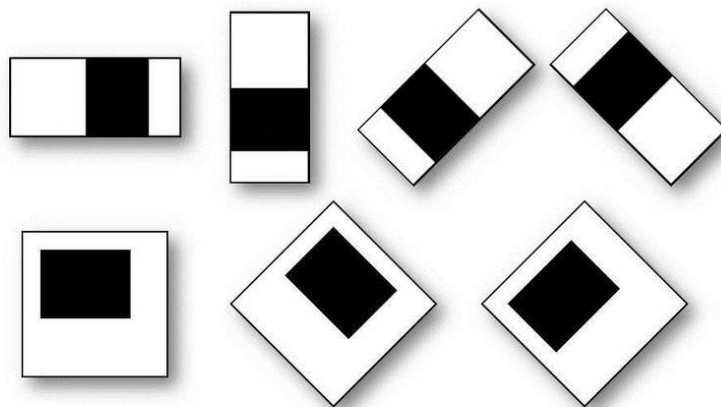


Рисунок 2.4 – Расширенные примитивы Хаара

Пример их наложения на изображение приведен на рисунке 2.5.



Рисунок 2.5 – Найденный примитив Хаара

В алгоритме Виолы-Джонса применяется бустер, чтобы уменьшить ошибки при определении лиц, то есть ложные срабатывания. Бустинг был описан во второй части второй главы, где было указано, что его суть заключается в последовательном обучении используемых классификаторов на основе ошибок предыдущих.

Каскадная модель алгоритма представляет собой дерево решений, состоящего из значений функции активации и общих условий перехода.[14]

На рисунке 2.6 представлен пример дерева решений, где $f(x)$ и $f_1(x)$ представляют собой булевы функции, а y – значения целевой функции.

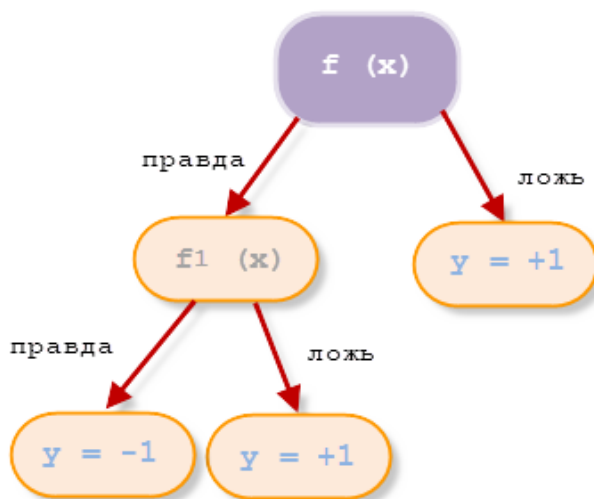


Рисунок 2.6 – Модель дерева решений

Все узлы дерева решений располагаются так, что каждый дальнейший узел содержит больше примитивов, чем предыдущий. Таким образом обработка изображения на начальных уровнях происходит на порядок быстрее, и чем

глубже обработка, тем больше времени требуется, однако на первых этапах получается быстро отсеять области, на которых не обнаружено необходимых образов. При малом количестве искомых объектов и с постепенным обучением всех каскадов время на обработку существенно сокращается.

На рисунке 2.7 представлена схема обработки, сложность которой рассчитывается как $O(x*y*z)$, где соответственно количества: x –этапов, y – примеры, z – признаки. При этом нужно учесть, что благодаря бустингу каждый следующий каскад к распознаванию подготовлен ещё лучше.

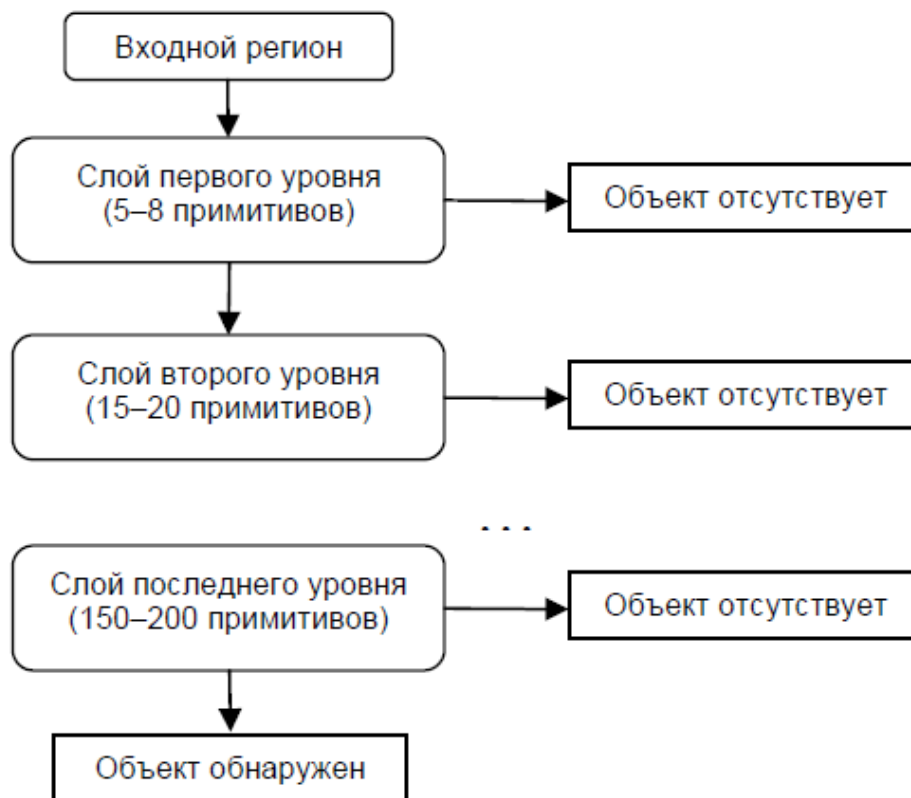


Рисунок 2.7 – Каскадная модель поиска

2.3.2 Алгоритм обработки

Учитывая все вышесказанное получим алгоритм обработки:

- а) выделить окна на изображении;
- б) произвести поиск первого классификатора, и если он не обнаружен, то происходит переход к пункту 3;
- в) произвести поиск второго, более усовершенствованного классификатора, и далее продолжать до последнего классификатора. Если не произведёт выход из цикла, значит, объект обнаружен;
- г) произвести сдвиг окна. Произвести изменение размера окна, после прохождения всего изображения.

2.3.3 Усовершенствование алгоритма

Метод Виолы-Джонса может быть успешно усовершенствован, так как для него можно использовать все методы предварительной подготовки входного изображения, основанные на теории обработки сигналов, что были рассмотрены предыдущих главах данной работы. Кроме того, часто применяется изменение изображения в полутоновый формат, чтобы исключить влияние на выбор цветов изображения.

В случаях работы на мощных ЭВМ возможна комбинация метода Виолы-Джонса и выделения специальных точек. Для обнаружения объекта используются точки, выбор и специфика которых определяется основными характеристиками объекта. Например, для лица такими точками могут выступать рот, глаза, нос. Выделенные паттерны проходят алгоритм Виолы-Джонса. Для ускорения данный метод применяется только для тех окон, где классификатором уже обнаружен объект. Кроме того, можно использовать итеративное углубление некоторых специальных точек, как, например, для глаз можно отдельно выделить зрачки и уголки глаз.

3 Разработка программного обеспечения и его настройка

На основе изученной и рассмотренной в предыдущих главах теоретической базы можно приступить к разработке собственного программного продукта. В качестве языка программирования выбран Python.

Python – высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Для удобства минимизирован синтаксис. [17]

3.1 Установка среды разработки

В качестве среды разработки была выбрана Microsoft Visual Studio, что представляет собой интегрированную среду разработки программного обеспечения и ряд иных инструментов. С их помощью можно разрабатывать как консольные приложения, так и приложения, имеющие графический интерфейс, веб-сайты, веб-приложения и другие программные продукты. Поддерживаются такие платформы, как Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework и Silverlight. Имеется соответствующая поддержка многих современных языков программирования, в том числе и Python.

Данная среда разработки является наиболее удобной и бесплатной, что немаловажно. Разработка производится на платформе Windows 10. Для установки Python в Microsoft Visual Studio необходимо указать все необходимые библиотеки и языки в программе установки. Соответствующие установки представлены на рисунке 3.1.

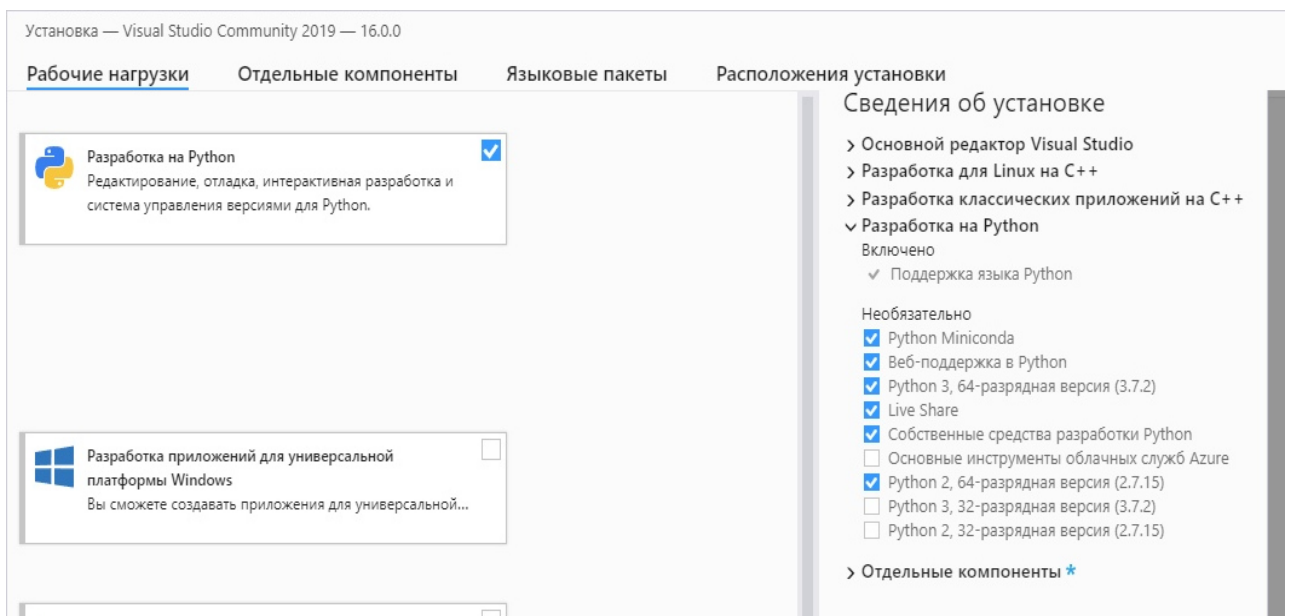


Рисунок 3.1 – Установка среды разработки

3.2 Установка библиотек

В качестве основной библиотеки была выбрана OpenCV.

OpenCV (OpenSourceComputerVision) – это библиотека, основным назначением которой является работа с изображениями, компьютерное зрение и обработка изображений и видео в реальном времени. Изначально была разработкой компании Intel, однако позднее в разработку включились WillowGarage, а ещё позднее Itseez, который после этого был приобретен Intel. Библиотека поставляется бесплатно и является кроссплатформенной.[18]

OpenCV написана на языке C++, хотя сохраняет интерфейс более старого языка C. Имеется привязка к Python, что является наиболее важным в рамках настоящей работы, но также есть привязки к Java и MATLAB / OCTAVE. API для различных интерфейсов можно найти в онлайн-источниках.

Официально проект OpenCV был запущен в 1999 году компанией IntelResearch. Приложения с данной библиотекой предполагали интенсивное использование процессора на такие функции, как трассировка лучей в реальном времени или стенды с трёхмерными дисплеями.

Первая альфа-версия была издана на конференции IEEE по распознаванию образов и в целом компьютерному зрению в 2000 году. Позднее с 2001 по 2005 годы были выпущены пять бета-версий. И только 2006 году была выпущена версия 1.0.

Вторая версия OpenCV 2 была выпущена в октябре 2009 года, она имела большие изменения в интерфейсе C++, целью которых было упрощение, включение новых функций и улучшение реализаций для уже существующих методов и функций. На данный момент официальные релизы происходят каждые полгода, а разработка ведётся независимой российской командой, которая поддерживается коммерческими организациями.

В 2012 году поддержку OpenCV взял на себя коммерческий фонд OpenCV.org. Он же производит поддержку пользовательского сайта данной библиотеки.

В 2016 году Intel выкупила Itseez, являющегося ведущим разработчиком библиотеки OpenCV.

Данная библиотека включает в себя работу с нейронными сетями, методы обработки изображений, высокоскоростное матричное вычисление, методы машинного зрения, отслеживание объектов в реальном времени, обработка трёхмерных изображений и другие.[19]

У библиотеки открытый код, а также множество доступных обучающих источников, которые полезны при выполнении данной работы.

Для установки данной библиотеки в Microsoft Visual Studio под Python, достаточно выполнить установку дополнительных пакетов через окружение Python.

На рисунке 3.2 представлено окно окружения Python, через которое производится установка библиотеки.[20]

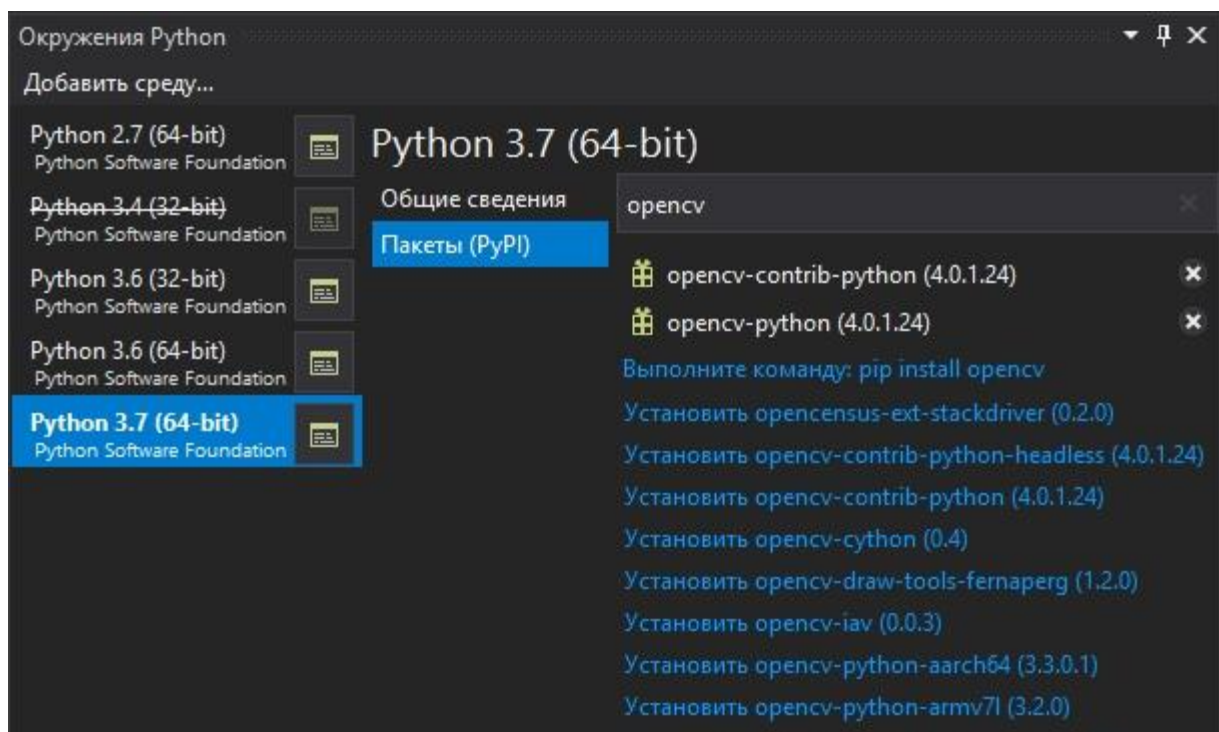


Рисунок 3.2 – Установка библиотеки OpenCV

3.3 Создание программы

Для удобства выполнения идентификации лиц код программы был разделён на два различных этапа:

- создание базы данных;
- идентификация входного изображения согласно базе данных.

3.3.1 Создание алгоритма и кода сохранения базы данных

На рисунке 3.3 представлены этапы создания базы данных.

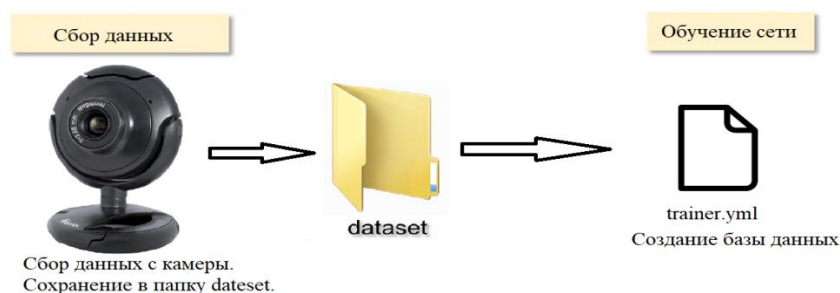


Рисунок 3.3 – Этапы создания базы данных

Для обучения нейронной сети на распознавание образов в первую очередь необходимо создать обучающую выборку, то есть сделать базу данных, состоящую из фотографий людей. В данном проекте идентификация человека будет производится по id-номеру. Поэтому обучающая выборка должна содержать по несколько фотографий одного человека для определения индивидуальных характеристик. Номер id вводится с клавиатуры.

Для поиска на изображении лица используются примитивы Хаара, которые входят в библиотеку OpenCV. После сохранения файла с проектом создания базы данных, необходимо переместить в ту же папку и файл, содержащий примитивы Хаара.

Рисунок 3.4 показывает содержание папки с проектом, где файл `Fase_Save_Base` – код программы, создающей обучающую выборку, и создающую базу данных в формате `.uml`, что представляет собой разработанный компанией Яндекс стандарт, используемый для принятия и сохранения данных в базе. Файлы `Fase_Detection` содержат данные по фазе детектирования и идентификации, которые будут рассмотрены далее.

компьютер > Рабочий стол > Fase > Fase_Save_Base >

Имя	Дата изменения	Тип	Размер
dataset	03.06.2019 0:52	Папка с файлами	
trainer	03.06.2019 0:36	Папка с файлами	
Fase_Detection	03.06.2019 0:52	Python source file	2 КБ
Fase_Detection	03.06.2019 0:25	Python Project	2 КБ
Fase_Detection.sln	03.06.2019 0:25	Visual Studio Solu...	1 КБ
Fase_Save_Base	03.06.2019 0:35	Python source file	3 КБ
Fase_Save_Base	03.06.2019 0:07	Python Project	2 КБ
Fase_Save_Base.sln	03.06.2019 0:07	Visual Studio Solu...	1 КБ
haarcascade_frontalface_default	11.04.2018 8:18	Документ XML	909 КБ

Рисунок 3.4 – Содержимое папки с проектом

На основе фотографий взятых с сайта университета была создана некоторая база данных, содержащая фотографии преподавательского состава кафедры «Электроника и робототехника». [21]

При создании базы для каждого преподавателя был введен id-номер, после чего программа определила область с лицом и сделала 30 снимков для каждого преподавателя. После создания фото, в базу данных сразу же сохранялись соответствующие материалы.

На рисунке 3.5 представлено содержимое папки dataset.



Рисунок 3.5 – Содержимое папки dataset

На рисунке 3.6 представлен процесс сохранения фото-файлов папку.

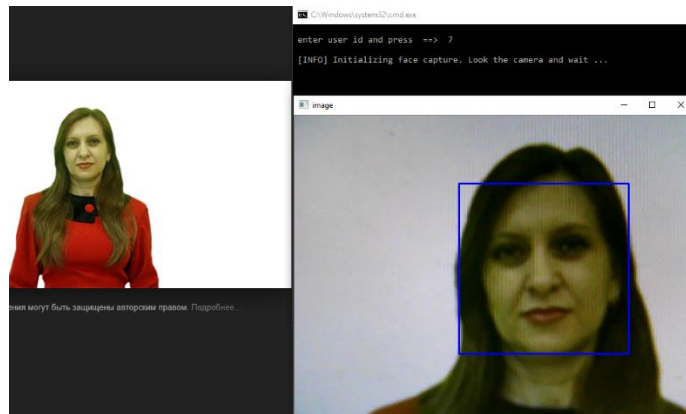


Рисунок 3.6 – Процесс работы программы №1

Код программы, создающей базу данных, представлен в Приложении А.

3.3.2 Создание программы идентифицирующей лица

На рисунке 3.7 представлены этапы распознавания лица.

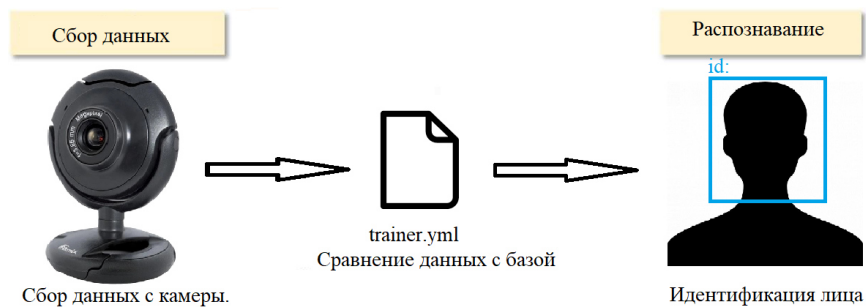


Рисунок 3.7 – Этапы распознавания лица

На рисунке 3.8 представлена структурная схема программы, распознающей лица.

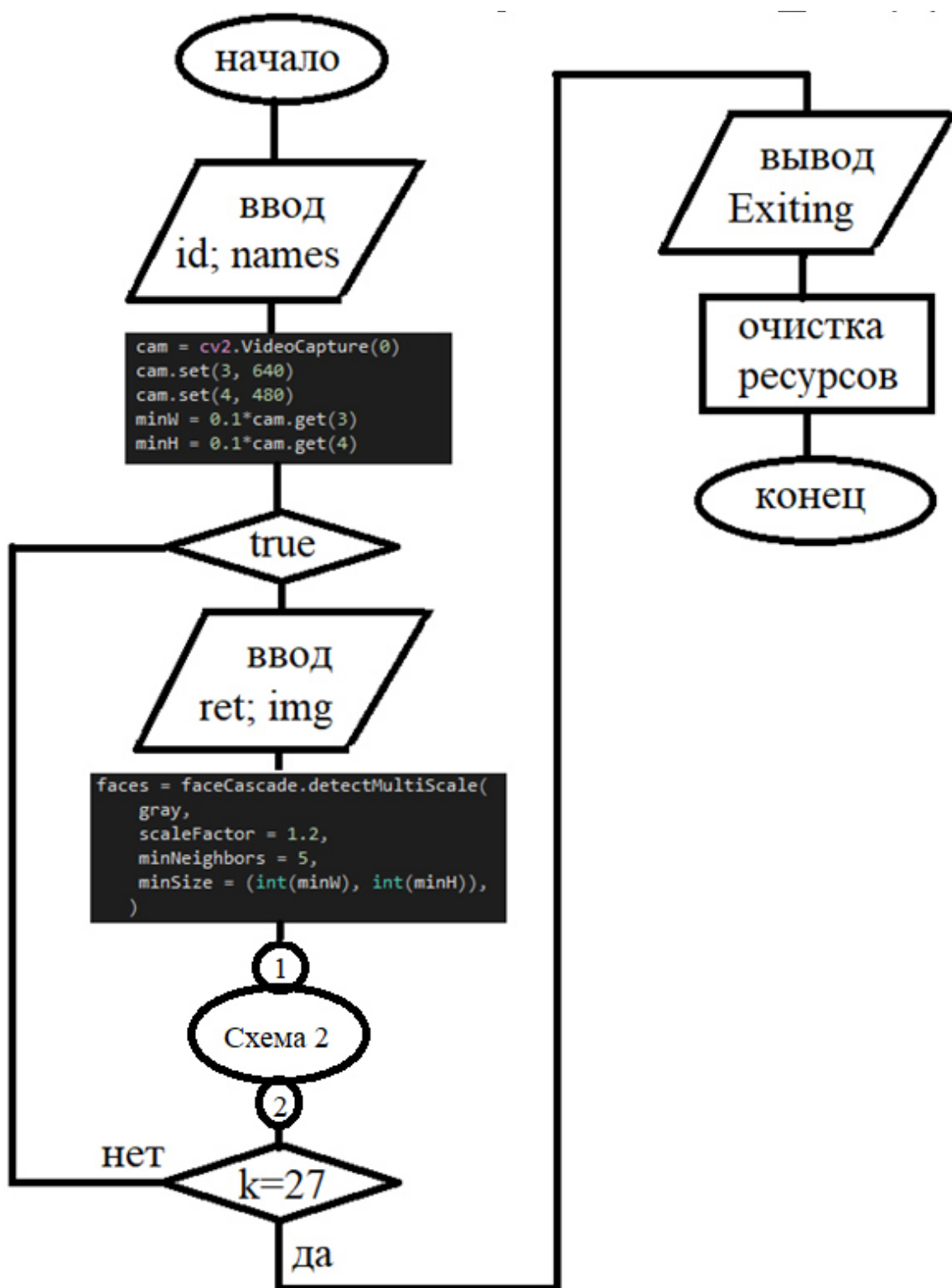


Рисунок 3.8 – Структурная схема-1 программы, идентифицирующей лица

На рисунке 3.9 представлена Схема 2, продолжение структурной схемы на рисунке 3.8.

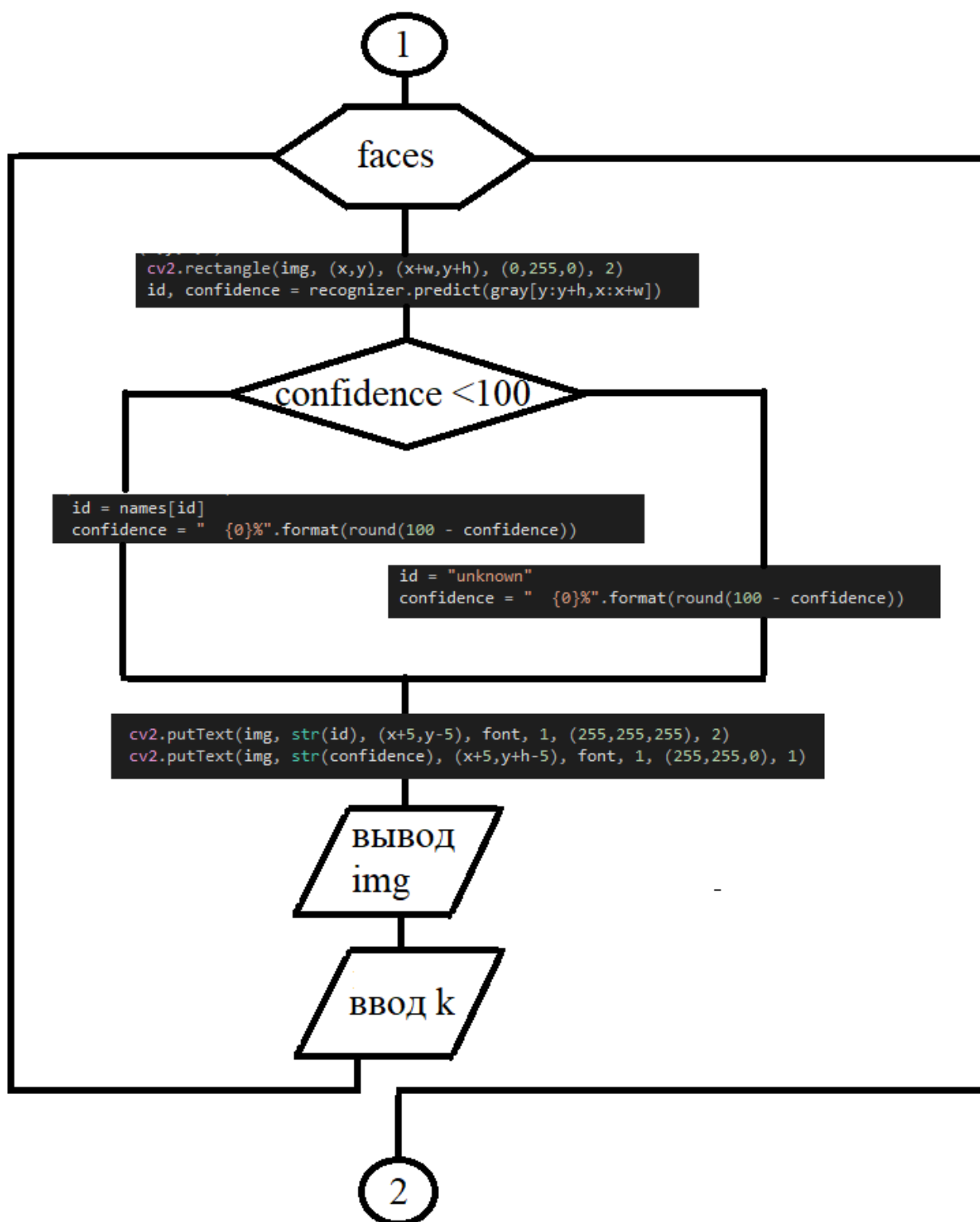


Рисунок 3.9 – Структурная схема-2, продолжение структурной схемы-1

Согласно структурной схеме была написана программа, код которой представлен в Приложении Б. Программа принимает входящий поток с веб-камеры и для начала ищет наличие лиц, как таковых. Если было обнаружено лицо, программа выделяет его в рамку и сверяет его с базой данных. Если в базе данных нашлось совпадение, то переменная `id` принимает значение, которое присвоено человеку с совпадением. Согласно значению `id` на экран будет выведено имя из вектора имён. При отсутствии совпадений и по умолчанию `id` имеет значение равное 0, и на экран будет выведено имя, стоящее первым в векторе имён, то есть «Неизвестно». Важно заметить, что обработка видеопотока ведётся покадрово, то есть каждый новый кадр будет обработан только тогда, когда окончена предыдущая итерация распознавания. На рисунке 3.10 представлен вектор имён.

```
names = ['None', 'It is Me', 'Balbaev', 'Auesova', 'Adambaev', 'Djeksenbinov', 'Baikenov', 'Golubeva']
```

Рисунок 3.10 – Вектор имён

3.3.3 Проверка работоспособности системы

Чтобы убедиться, что программа работает корректно необходимо провести экспериментальные исследования. На рисунке 3.11 – 3.12 представлены ошибки и некорректные результаты работы.

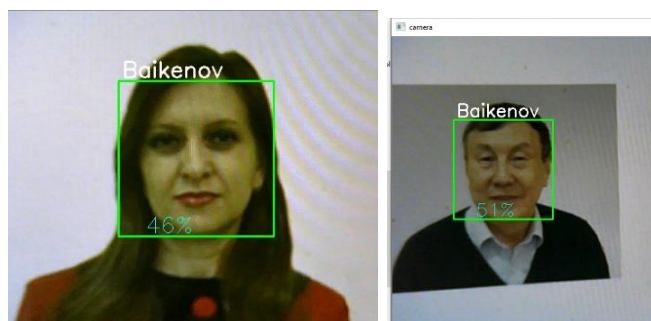


Рисунок 3.11 – Ошибочное определение

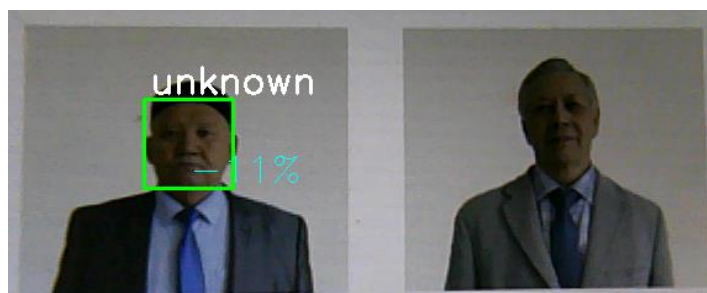


Рисунок 3.12 – Несрабатывание системы

Вероятной причиной ошибок является недостаточное обучение системы, ввиду нехватки обучающего материала, или же неточности фотографии, ввиду выполнения фотографии с фотографии.

При увеличении количества фотографий до 40, а также их хорошего максимально возможного повышения их качества система справляется намного лучше. Результаты представлены на рисунках 3.13-3.14.

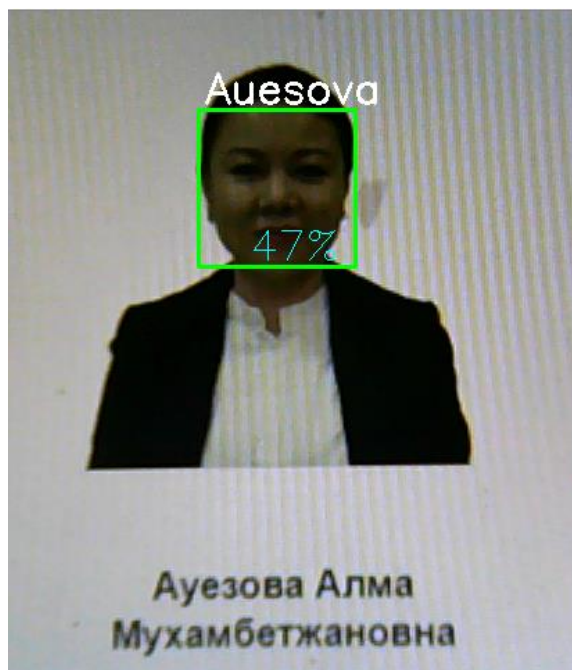


Рисунок 3.13 – Корректная идентификация

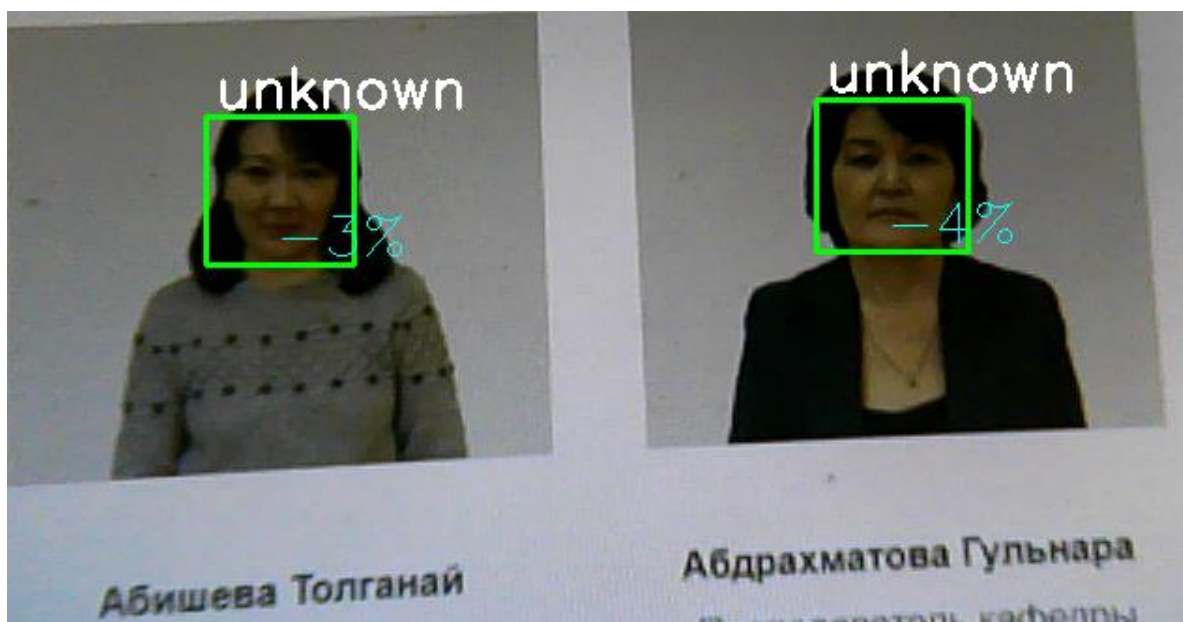


Рисунок 3.14 – Корректное распознавание

На рисунке 3.15 представлено исправление ошибки.

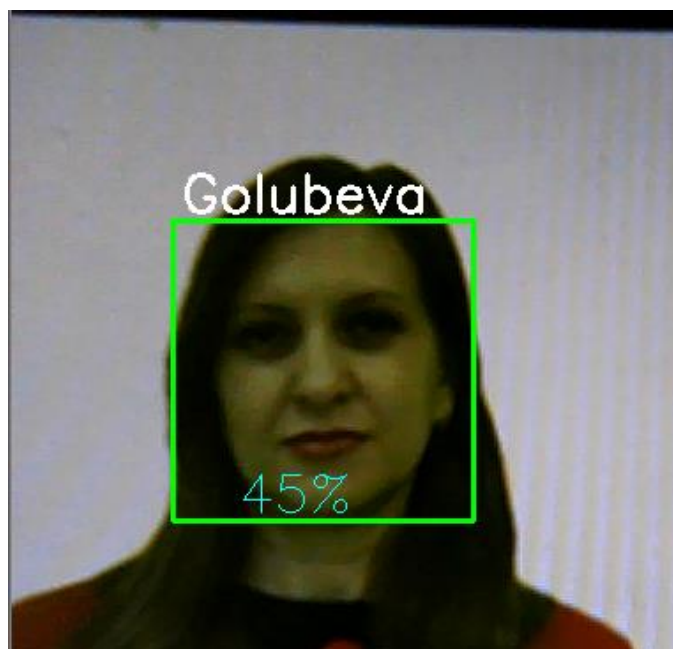


Рисунок 3.15 – Исправление ошибки

На рисунке 3.16 представлены оставшиеся преподаватели.

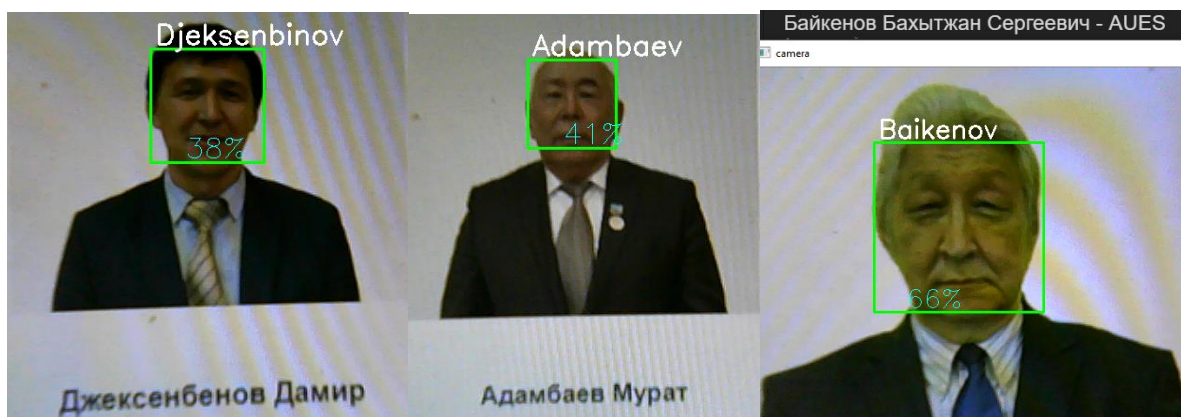


Рисунок 3.16 – Результат идентификации

3.4 Запуск программы на Raspberry Pi

Raspberry Pi — одноплатный малого размера, изначально разработанный для обучения информатике на основе данной бюджетной системы, однако позже получивший более широкое применение. Основным разработчиком является Raspberry Pi Foundation. Всего за пять лет было продано более 12,5 миллионов подобных устройств.

Предполагается, что операционная система Raspbian уже установлена на Raspberry Pi, этапы установки операционной системы и вспомогательных элементов в данной работе не рассматриваются.

Raspbian — это операционная система, основу которой составляет Debian. На сегодняшний день существует несколько версий Raspbian, такие как, например, Raspbian Stretch и Raspbian Jessie. Официально Raspbian представлена с 2015 года Raspberry Pi Foundation как основная операционная система Raspberry Pi. Raspbian была создана Майком Томпсоном и Питером Грином как самостоятельный и независимый проект. Первоначальная сборка была выполнена в июне 2012 года. Данная операционная система активно разрабатывается и усовершенствуется. Raspbian является максимально оптимизированной под процессоры ARM, обладающие низкой производительностью и применяемые в Raspberry Pi.

Для запуска программного продукта данной дипломной работы необходима установка библиотеки OpenCV на Raspberry Pi. Для этого необходимо в командной строке ввести следующие команды, представленные на рисунке 3.17. Это займёт продолжительное время, однако после выполнения установки можно будет сразу же запустить написанную ранее программу.

```
# ==== Необязательно ====
# Для тех, у кого мало места на флешке
sudo apt-get purge wolfram-engine
sudo apt-get purge libreoffice*
sudo apt-get clean
sudo apt-get autoremove
# =====

sudo apt-get -y update && sudo apt-get -y dist-upgrade

# sudo apt-get install -y build-essential pkg-config # Уже предустановлены в системе
sudo apt-get install -y cmake cmake-curses-gui libgtk2.0-dev
sudo apt-get install -y libavcodec-dev libavformat-dev libswscale-dev libv4l-dev libx264-dev libxvidcore-dev
sudo apt-get install -y libjpeg-dev libpng12-dev libtiff5-dev libjasper-dev

# ==== Необязательно ====
sudo apt-get install -y gfortran libatlas-base-dev
sudo apt-get install -y libdc1394-22-dev libavresample-dev libgphoto2-dev ffmpeg libgtk-3-dev
sudo apt-get install -y libvtk6-dev
# =====

sudo pip3 install numpy

cd /home/pi
mkdir opencv
cd opencv

wget https://github.com/opencv/opencv/archive/3.4.1.zip -O opencv_source.zip
wget https://github.com/opencv/opencv_contrib/archive/3.4.1.zip -O opencv_contrib.zip
unzip opencv_source.zip
unzip opencv_contrib.zip

rm opencv_source.zip
rm opencv_contrib.zip

cd /home/pi/opencv/opencv-3.4.1
mkdir build
cd build

# Если мало места, то убрать параметр OPENCV_EXTRA_MODULES_PATH
cmake -D CMAKE_BUILD_TYPE=RELEASE \
      -D CMAKE_INSTALL_PREFIX=/usr/local \
      -D INSTALL_C_EXAMPLES=OFF \
      -D INSTALL_PYTHON_EXAMPLES=ON \
      -D OPENCV_EXTRA_MODULES_PATH=/home/pi/opencv/opencv_contrib-3.4.1/modules \
      -D BUILD_EXAMPLES=ON \
      -D BUILD_DOCS=ON \
      -D ENABLE_NEON=ON ..

make -j1

sudo make install

sudo ldconfig
```

Рисунок 3.17 – Команды для установки библиотеки

4 Экономическая эффективность

Данный проект представляет собой реализацию программного обеспечения, распознающего лица. Программа может использоваться в качестве проверки личности при доступе к данным или в целях проверки подлинности документов, то есть их соответствие реальному лицу. Для достижения поставленной задачи был произведён анализ алгоритмов идентификации, сред разработки и вспомогательных библиотек.

В данной части дипломной работы представлен расчет цены полученного программного обеспечения, себестоимости продукта, а также всех её составляющих. Поэтапно произведён расчет трудоёмкости, заработной платы участников проекта, материальных затрат на проект, затраты на коммунальные услуги и аренду помещения и амортизационные затраты. На основе данных расчётов была получена себестоимость разработанного программного обеспечения. В конце приведён расчет цены проекта.[22]

4.1 Расчёт трудоёмкости разработки программного продукта

Трудоёмкость определяется временем, потраченным на выполнение работы. Для определения трудоёмкости при реализации проекта необходимо определить все виды работ, выполненных при подготовке и в процессе работы. Для качественного выполнения работы в первую очередь необходимо определить последовательность всех этапов, при этом учесть те работы, которые возможно выполнять параллельно, для экономии времени работ. В процессе выполнения работы были выделены следующие виды работ: разработка технического задания, анализ технического задания и теоретическое исследование, разработка алгоритма и построение блок-схемы, на основе которых будет следующим этапом составлена программа, проверка работоспособности программного обеспечения и корректировка ошибок, а также оформление результатов работы в виде соответствующей документации. Также следует учесть, что в выполнении данной работы участвовал научный руководитель проекта, то есть подсчет времени необходимо произвести для каждого участника проекта отдельно.

Ниже представлена формула(4.1) для подсчета трудоёмкости, чел.ч:

$$t_{\text{тр}} = t_{\text{оп}} + t_{\text{бс}} + t_{\text{ал}} + t_{\text{пр}} + t_{\text{отл}} + t_{\text{док}}, \quad (4.1)$$

где: $t_{\text{оп}}$ – время на подготовку и постановку задачи, чел.ч;

$t_{\text{бс}}$ – разработка блок-схемы для написания кода программы, чел.ч;

$t_{\text{ал}}$ – время на создание алгоритма программы, чел.ч;

$t_{\text{пр}}$ – время на написание кода программы, чел.ч;

$t_{\text{отл}}$ – отладка программы на Raspberry Pi, чел.ч;

$t_{\text{док}}$ – подготовка документации, чел.ч.

Соответствующие значения времени, затраченные при выполнении работы, представлены в таблице 4.1. Часы переведены в дни и округлены по избытку.

Таблица 4.1 – Трудоёмкость каждого этапа работы, чел.ч

Этап разработки ПП	Вид работы на данном этапе	Трудоёмкость разработки ПП, чел*ч	
		руководитель	разработчик
Разработка ТЗ		1	2
Теоретическое исследование	Обзор литературы по теме	5	32
	Выбор среды разработки и сторонних библиотек	3	24
Разработка приложения	Разработка ПП	10	64
Тестирование ПП	Тестирование ПП	16	56
Оформление пояснительной записки		-	54
Оформление иллюстративного материала		-	10
Итого часов		35	242
Итого дней(по 8 часов в день)		5	31

4.2 Заработная плата

После определения всех трудозатрат можно приступить к расчёту заработной платы.

Фонд оплаты труда рассчитывается по формуле (4.2):

$$З_{\text{фот}} = З_{\text{осн}} + З_{\text{доп}}, \quad (4.2)$$

где: $З_{\text{осн}}$ – основная заработная плата, тенге;

$З_{\text{доп.}}$ – дополнительная заработная плата, тенге.

Соответственно $З_{\text{доп.}}$ находится как 20% от основной заработной платы:

$$З_{\text{доп}} = З_{\text{осн}} \cdot 0.2. \quad (4.3)$$

Основная заработная плата рассчитывается по формуле (4.4), где $t_{\text{тр}}$ – трудоемкость (чел/дни):

$$З_{\text{осн}} = \sum_{i=1}^n t_{\text{тр}.i} \cdot З_{\text{дн}.i}, \quad (4.4)$$

В формуле (4.4) $Z_{\text{дн}}$ – это заработная оплата труда в день, тенге.

Для того, чтобы найти оплату труда за день, необходимо учесть месячный оклад специалиста и количество рабочих дней в месяце. Согласно рисунку 4.1 среднее количество дней при пятидневной рабочей неделе в 8-мичасовом рабочем дне в 2019 году 20.5, для удобства расчетов возьмём по избытку и округлим до 21. [23]

Месяц, квартал, полугодие 9 месяцев, год	Количество дней		Рабочие дни							
	Календарные дни	Календарные дни без учета праздничных дней	40-часовая рабочая неделя				36-часовая рабочая неделя			
			График работы				График работы			
			с 1-го числа				с 1-го числа			
			до конца месяца				до конца месяца			
			пятидневка		шестидневка		пятидневка		шестидневка	
	дни		дни	часы	дни	часы	дни	часы	дни	часы
Январь	31	29	20	160	24	160	20	144,00	24	144,00
Февраль	28	28	20	160	24	160	20	144,00	24	144,00
Март*	31	27	17	136	22	146	17	122,40	22	132,00
I квартал	90	84	57	456,00	70	466	57	410,40	70	420
Апрель	30	30	22	176	26	174	22	158,40	26	156,00
Май	31	28	20	160	24	160	20	144,00	24	144,00
Июнь	30	30	20	160	25	165	20	144,00	25	150,00
II квартал	91	88	62	496,00	75	499,0	62	446,4	75	450,0
I полугодие	181	172	119	952,00	145	965,0	119	856,8	145	870,0
Июль*	31	30	22	176	26	176	22	158,40	26	156,00
Август	31	30	21	168	26	172	21	151,20	26	156,00
Сентябрь	30	30	21	168	25	167	21	151,20	25	150,00
III квартал	92	90	64	512,00	77	515,0	64	460,8	77	462,0
9 месяцев	273	262	183	1464,00	222	1480	183	1317,6	222	1332
Октябрь	31	31	23	184	27	181	23	165,60	27	162,00
Ноябрь	30	30	21	168	26	172	21	151,20	26	156,00
Декабрь*	31	28	19	152	23	153	19	136,80	23	138,00
IV квартал	92	89	63	504,00	76	506,0	63	453,6	76	456,0
Год	365	351	246	1968,00	298	1986	246	1771,2	298	1788
Среднемесячный баланс		29,25	20,50	164,00	24,83	165,50	20,50	147,60	24,83	149,00

* При совпадении праздничных дней с выходными выходным днем является следующий после праздничного рабочий день.

Рисунок 4.1 – Баланс рабочего времени на 2019 год

Согласно компании интернет-рекрутмента HeadHunter заработная плата программиста на языке Python составляет в среднем 260 000 тенге. Оплата труда руководителя проекта оценивается в среднем в 350 000 тенге.[24]

Для нахождения дневной оплаты труда необходимо среднюю заработную плату разделить на среднее количество рабочих дней. Тогда заработная плата каждого из работников:

$$Z_{\text{дн1}} = \frac{265\,000}{21} = 12\,619 \text{ тенге,} \quad (4.5)$$

$$Z_{\text{дн2}} = \frac{350\,000}{21} = 16\,667 \text{ тенге.} \quad (4.6)$$

Теперь рассчитаем основную заработную плату для каждого работника по формуле (4.4) и общую:

$$З_{осн1} = 31 \cdot 12\,619 = 391\,189 \text{ тенге,}$$

$$З_{осн2} = 5 \cdot 16\,667 = 83\,335 \text{ тенге,}$$

$$З_{осн} = З_{осн1} + З_{осн2} = 474\,524 \text{ тенге} \quad (4.7)$$

Тогда дополнительная заработная плата будет составлять:

$$З_{доп} = З_{осн} \cdot 0.2 = 94\,905 \text{ тенге.}$$

Подставив полученные значения в формулу (4.2) для расчёта фонда оплаты труда, получим общую сумму:

$$З_{фот} = 474\,524 + 94\,905 = 569\,429 \text{ тенге.}$$

Социальный налог в Республике Казахстан на момент 2019 года составляет 9,5% от дохода разработчика, однако следует учесть, что пенсионные отчисления данным налогом не облагаются, тогда формула (4.8) имеет вид:

$$О_{осн} = (З_{фот} - З_{по}) \cdot 0.095, \quad (4.8)$$

где: $З_{по}$ – пенсионные отчисления.

Пенсионные отчисления можно рассчитать как 10% от заработной платы, по формуле:

$$З_{по} = З_{фот} \cdot 0.1 = 569\,429 \cdot 0.1 = 56\,943 \text{ тенге.} \quad (4.9)$$

Подставив значения в формулу (4.8), найдем социальный налог:

$$О_{осн} = (569\,429 - 56\,943) \cdot 0.095 = 48\,687.$$

4.3 Материальные расходы

Расчёт капиталовложений на проект можно найти как сумму капиталовложений на реализацию проекта и капиталовложений на оборудование. В качестве оборудования в данном случае выступает Raspberry Pi, его периферия, то есть мышь, клавиатура и камера.

Следует отметить, что программное обеспечение, используемое в ходе выполнения работы, а именно операционная система для микрокомпьютера Raspbean, является бесплатной.

Формула (4.10) капиталовложений имеет вид:

$$K_{\text{пр}} = K_{\text{по}} + K_{\text{об}}, \quad (4.10)$$

где: $K_{\text{пр}}$ – капиталовложения необходимые на проект;

$K_{\text{об}}$ – капиталовложения на оборудование;

$K_{\text{по}}$ – капиталовложения на программное обеспечение, которая равна нулю.

Все основные материальные затраты представлены в таблице 4.2.

Таблица 4.2 – Основные материальные затраты

Ресурс	Количество	Цена	Сумма
Raspberry Pi	1	30000	30000
Web-Камера, Genius	1	6500	6500
Мышь, USB	1	1000	1000
Клавиатура, Jedel, USB	1	1500	1500
Программное обеспечение			0
			39000
Транспортные затраты	5%		1950
Сумма			40950

Также следует учесть дополнительные затраты на канцелярию и иные ресурсы, использованные в ходе разработки и оформления, такие как канцелярские изделия и прочее. Дополнительные расходы представлены в таблице 4.3.

Таблица 4.3 – Дополнительные расходы

Ресурс	Количество	Цена	Сумма
Бумага для оргтехники, пачка	1	1380	1380
Картридж для принтера	1	5000	5000
Канцелярские товары		1000	1000
Сумма			7380

Тогда суммарные материальные расходы будут составлять:

$$M = K_{\text{пр}} + K_{\text{доп}} = 40950 + 7380 = 48\,330. \quad (4.11)$$

4.4 Расчет электроэнергии

Так как все работы производились на электрической технике, то необходимо произвести расчет затрат на электроэнергию с учётом общей мощности всех приборов. Для этого воспользуемся формулой (4.12):

$$P_{\Sigma} = M \cdot k_3 \cdot T \cdot C_{\text{кВт-ч}}, \quad (4.12)$$

где: M – мощность ЭВМ, кВт;

k_3 – коэффициент загрузки (0.8);

$C_{\text{кВт-ч}}$ – стоимость 1 кВт·ч электроэнергии, тенге/ кВт·ч;

T – время работы, час (344 ч.).

В таблице 4.4 показаны значения для расчета затрат на электроэнергию.

Таблица 4.4 - Затраты на электроэнергию

Наименование оборудования	Паспортная мощность, кВт	Коэффициент загрузки	Время работы оборудования для разработки ПП, час	Цена эл/э, тенге/кВт-час;	Сумма, тенге
Компьютер	0,15	0,8	172	16,53	341,17
Принтер	0,1	0,8	5	16,53	6,612
Монитор	0,15	0,8	242	16,53	480,03
Raspberry Pi	0,015	0,8	70	16,53	13,8852
ИТОГО					841,7

Произведём расчёт, подставив значения в формулу (4.12) для расчета расходов на электроэнергию:

$$P_{\Sigma 1} = 0.15 \cdot 0.8 \cdot 172 \cdot 16,53 = 341,17 \text{ тенге,}$$

$$P_{\Sigma 2} = 0.1 \cdot 0.8 \cdot 5 \cdot 16,53 = 6,6 \text{ тенге,}$$

$$P_{\Sigma 3} = 0.1 \cdot 0.8 \cdot 242 \cdot 16,53 = 480,03 \text{ тенге,}$$

$$P_{\Sigma 4} = 0.1 \cdot 0.8 \cdot 70 \cdot 16,53 = 13,9 \text{ тенге,}$$

$$P_{\Sigma} = P_{\Sigma 1} + P_{\Sigma 2} + P_{\Sigma 3} + P_{\Sigma 4} = 841,7 \text{ тенге.} \quad (4.13)$$

4.5 Амортизационные расходы

Любая техника со временем устаревает и нуждается в обновлении. Также не исключены поломки. Затраты на обновление и ремонт техники необходимо учесть как амортизационные отчисления. Они зависят от стоимости техники, используемой в разработке программного обеспечения.

Для расчета амортизационных отчислений используется формула (4.14):

$$P_{\text{ам}} = \frac{C_{\text{обор}} \cdot N_a \cdot N}{100 \cdot 12 \cdot t}. \quad (4.14)$$

где: N_a – норма амортизации, составляющая 25%;

$C_{\text{обор}}$ – первоначальная стоимость техники, тенге;

N – время использования персональной техники;

t – количество рабочих дней в месяце, дни.

Для выполнения данной работы был использован персональный компьютер стоимостью 200 000 тенге, монитор Acer в стоимость 70 000 тенге, принтер HP LaserJet 1018 стоимостью 25 000 тенге. Реализация продукта происходит на Raspberry Pi, но расчет амортизации на него не производится так как он является расходным материалом и входит в стоимость программного продукта. Также не происходит расчёта для программного обеспечения, используемого в процессе разработки, так как оно является бесплатным.

Рассчитаем по формуле (4.14) расходы на амортизацию для каждого из них отдельно, а затем их сумму:

$$P_{\text{ам1}} = \frac{200\,000 \cdot 25 \cdot 21,5}{100 \cdot 12 \cdot 20} = 4479,16 \text{ тенге,}$$

$$P_{\text{ам2}} = \frac{25\,000 \cdot 25 \cdot 0,625}{100 \cdot 12 \cdot 20} = 16,28 \text{ тенге,}$$

$$P_{\text{ам4}} = \frac{70\,000 \cdot 25 \cdot 30,25}{100 \cdot 12 \cdot 20} = 2205,72 \text{ тенге,}$$

$$P_{\text{ам}} = P_{\text{ам1}} + P_{\text{ам2}} + P_{\text{ам3}} + P_{\text{ам4}} = 6701,16 \text{ тенге.} \quad (4.15)$$

4.6 Накладные расходы

К накладным расходам относятся арендная плата за помещение и стоимость его обслуживания, оплату информационно-коммуникационных услуг и т.д. Так как суммарное время работы 31 день, то все расчеты можно округлить до расчета для одного месяца.

Весь список накладных расходов представлен в таблице 4.5.

Таблица 4.5 – Накладные расходы

Наименование материального ресурса	Количество	Цена за единицу, тг	Сумма, тг
Услуги интернет провайдера	1 месяц	4000	4000
Аренда рабочего помещения	10 кв. м	5000	50000
Обслуживание помещения		1000	1000
Система пожарной сигнализации	1 месяц	12000	12000
Охрана помещения	1 месяц	9000	9000
Сумма			76000

4.7 Себестоимость продукции

Расчет $C_{пп}$, то есть себестоимости программного продукта, полученного в результате выполнения данного проекта, можно получить как сумму всех соответствующих затрат на его реализацию, а также оплаты труда работников. Данный расчёт выполнен по формуле (4.16):

$$C_{пп} = Z_{\text{фот}} + O_{\text{осн}} + M + P_{\text{ам}} + P_{\text{эл}} + P_{\text{нак}}, \quad (4.16)$$

где: $Z_{\text{фот}}$ – фонд оплаты труда, предназначенный для разработчиков, тенге;

$O_{\text{осн}}$ – отчисления по социальному налогу, тенге;

M – затраты на материалы, тенге;

$P_{\text{ам}}$ – расходы на амортизацию техники, тенге;

$P_{\text{эл}}$ – расходы на электроэнергию, тенге;

$P_{\text{нак}}$ – накладные расходы, тенге.

Подставив значения в формулу (4.16), посчитанные ранее и округлённые, получим:

$$C_{пп} = 569\,429 + 48\,687 + 48\,330 + 6\,701,16 + 841,7 + 76\,000 = 749\,989.$$

4.8 Цена реализации программного продукта

Найти цену реализации программного обеспечения можно с помощью следующей формулы (4.17):

$$C_p = C_d + C_d * \text{НДС}, \quad (4.17)$$

где: C_d – договорная цена.

Налог на добавленную стоимость(НДС), размер которого регулируется законодательно, согласно Налоговому Кодексу РК на 2019 год составляет 12%.

Размер договорной цены программного продукта устанавливается с учетом качества, эффективности и сроков ее выполнения на уровне, отвечающем экономическим интересам заказчика и исполнителя. Договорная цена (Цд) для прикладных ПП рассчитывается по формуле (4.18):

$$Цд = С_{пп}(1 + \frac{Р}{100}), \quad (4.18)$$

где: $C_{пп}$ - затраты на весь проект, тенге;

P - средний уровень рентабельности ПП, % (принимается в размере 20-30%).

Сделаем расчет договорной цены по формуле (4.18), указанной выше:

$$Цд = 749\,989 * \left(1 + \frac{25}{100}\right) = 937\,487 \text{ тенге.}$$

Теперь по формуле Цр можно сделать соответствующий расчет цены на реализацию данного программного продукта.

Подставим значения в формулу (4.17):

$$Цр = 937\,487 + 937\,487 * 0.12 = 1\,049\,986 \text{ тенге.}$$

Вывод: на основе установленных вначале пунктов и их последовательного выполнения был создан программный продукт, для которого произведёт соответствующий экономический расчет. Цена реализации полученного в ходе работы программного продукта составляет 1 049 986 тенге.

4.9 Оценка социально-экономических результатов функционирования программного продукта

Для оценки социально-экономической эффективности проекта необходимо представить области, в которых данный проект применяется, а затем рассчитать период, за который прибыль или экономия от применения данного прибора достигнет размера затраченных на него средств. Соответственно, чем ниже коэффициент срока окупаемости, тем более выгодным является проект.

Предполагается использовать данный проект в области безопасности для сравнения подлинности документов ввиду того, что всё большее количество людей стремятся покинуть страну, используя чужие документы. А также идентификация людей, скрывающихся от уплаты налогов или находящихся в розыске, при попытке пересечь границу страны.

Нелегальная миграция работоспособного населения приводит к появлению на рабочих местах неквалифицированных работников. Согласно анализу миграционной подвижности в Республике Казахстан большая часть мигрантов имеет низкий уровень образования.[25] Это в первую очередь влияет на среднюю оплату труда. Но также негативно влияет на заинтересованность предпринимателей в развитии производственного процесса. Все это может приводить к коррупции, нелегальной занятости и укрыванию доходов. В качестве пользы от незаконной миграции можно указать заполнение рынка труда по вакансиям, невостребованным гражданами самой принимающей страны. Незаконная миграция способна также влиять и на экономическую безопасность страны.

При этом выделяют две основные экономические угрозы:

а) преднамеренные - то есть целенаправленное нелегальное трудоустройство мигрантов;

б) непреднамеренные угрозы – деятельность самих мигрантов и социальных структур, являющаяся причиной опасных деяний мигрантов.

Ещё одним влиянием на экономику является теневой бизнес, и отсутствие должных отчислений в бюджет государства, что приводит к росту экономики соседних стран, и снижению экономики собственной экономики страны.

Согласно Всемирному банку ВВП Таджикистана состоит на 45% из переводов мигрантов в другие страны, также и 38% ВВП Молдавии. При этом более 40% из всемирного объема финансовых переводов осуществляется мигрантами нелегально.[26-28]

Таким образом важно пресекать любую незаконную деятельность, в том числе и миграцию, как из страны, так и внутрь страны, ведь от этого зависит непосредственно экономическая безопасность страны.

Полученный в результате программный продукт при внедрении его в государственные структуры позволит создать базу данных лиц мигрантов и далее отслеживать их деятельность, что согласуется с Концепцией миграционной политики Республики Казахстан на 2007-2015 годы. А именно регулирование незаконной миграции: важной чертой миграционной политики Республики Казахстан будут являться обеспечение безопасности страны, борьба с незаконной миграцией. В этом направлении будут технически и организационно по-новому укрепляться, и переоборудоваться границы страны, совершенствоваться система технического оформления и предоставления виз. В целях соблюдения принципов транспарентности и легитимности будет осуществляться на постоянной основе мониторинг мигрантов в части их законного допуска и добросовестного пребывания в стране.[29]

Для получения разрешения на работу мигрант должен внести в казну сумму в размере 2 месячных расчетных показателей, на момент 2019 года 1МРП составляет 2525. Также трудовые мигранты выплачивают налоги по налоговой

ставке, предусмотренной статьями 154-1, 192, 194 и 201 Налогового кодекса РК в бюджет РК. При минимальной заработной плате, установленной законом от 30 ноября 2018 года «О республиканском бюджете на 2019-2021 год» в размере 42 500, в казну государства должны поступать ежемесячно налоги по налоговой ставке 20%.[30-31]

Тогда ежемесячно от каждого мигранта в казну должна поступать сумма:

$$N_1 = 2 * 2525 + 42500 * 0.2 = 13550 \text{ тенге.} \quad (4.19)$$

За 2018 год более 2 тысяч работодателей получили штрафы, за использование нелегальной рабочей силы из других стран. Если даже считать минимально, что каждый из них нанял по одному человеку, то получается сумма, во много раз превышающая стоимость моего проекта.

$$N_2 = 13550 * 2000 = 27\,100\,000 \text{ тенге.} \quad (4.20)$$

Тогда как для возмещения затрат на приобретение представленного проекта достаточно выявление меньшего количества правонарушителей, что показывает эффективность проекта. А именно:

$$N_q = \frac{1\,049\,986}{13550} = 78. \quad (4.21)$$

5 Безопасность жизнедеятельности

5.1 Теоретические основы безопасности жизнедеятельности

Безопасность жизнедеятельности (БЖД) – наука, исследующая законы возникновения опасных ситуаций, мерах их предупреждения и минимизации влияния на человека. Данная учебная дисциплина состоит из таких частей, как охрана труда, то есть безопасность на производстве, план мероприятий в случае чрезвычайных ситуаций и охрана окружающей среды.[32]

Основной целью БЖД является максимальная адаптация человека в эргатических системах, то есть системах, в которых человек принимает активное участие, при условии сохранения его здоровья и высокой работоспособности. К эрготическим системам можно отнести системы типа «человек-машина», «человек-машина-окружающая среда» и подобные им.

Любая деятельность человека может быть потенциально опасной. При этом возникает при сочетании нескольких условий, в следствии чего человеку может быть нанесен какой-либо вред.

Опасная ситуация – это момент времени, обладающий достаточно большой вероятностью возникновения несчастного случая.

Опасность – ситуация, ведущая к возникновению процессов и явлений, воздействие которых на человека пагубно, а также которые могут привести к материальному ущербу или влияющих негативно на окружающую среду.

Опасность – это центральное понятие БЖД, она присутствует в любой человеческой деятельности в той или иной мере.

Источниками опасностей могут выступать:

- а) сам человек, как сложная система, организм, который может обладать физиологическими и наследственными негативными явлениями;
- б) ограниченность возможностей организма, антропометрические и психологические показатели каждого человека, которые могут быть неподходящими при определённой деятельности;
- в) среда обитания и её элементы, такие как продукты труда, предметы и средства, энергия, используемые в процессе деятельности, климатические факторы;
- г) условия жизни и условия труда, к которым относятся влажность и температура, освещение, животный и растительный мир;
- д) каждый человек в частности так же, как и коллектив людей;
- е) процессы взаимодействия людей со средой обитания.

Так как опасности имеют множество факторов, которые могут накладываться друг на друга, то важно их классифицировать и систематизировать для лучшего понимания природы самих опасностей.

Таксономия – наука, исследующая, классифицирующая и систематизирующая сложные явления, имеющие множество факторов.

5.2 Вредные факторы на производстве

Тема настоящей работы связана с программированием и работой с компьютером. Таким образом наиболее важным является создание благоприятной среды для работы человека, а также правильная организация рабочего пространства.

Для программиста в первую очередь важна посадка за компьютером, ввиду долгой и монотонной работы. В противном случае есть риск возникновения заболеваний опорно-двигательного аппарата и зрения. На рисунке 5.1 представлена схема организации рабочего пространства для работника с ЭВМ.[33]

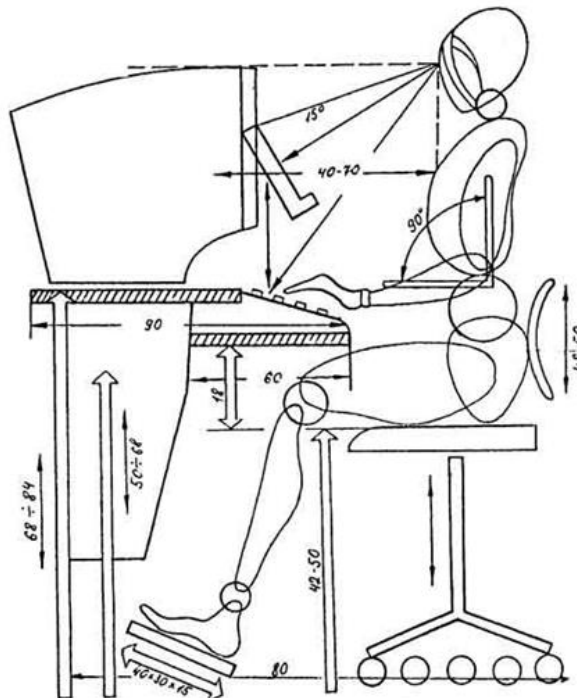


Рисунок 5.1 – Схема организации рабочего пространства

К основным вредные факторы, влияющие на программиста, подразделяются на:

а) физические, к которым относятся повышенные уровни электромагнитного излучения, рентгеновского, ультрафиолетового и инфракрасного излучения. Так как ЭВМ является электрической машиной, опасность представляет статическое электричество, важно обратить внимание на повышенное значение напряжения в электрической цепи, замыкание которой может произойти через тело человека. Повышенный уровень шума, а также его непрерывность. Ввиду работы с монитором вред может нанести повышенный уровень прямой и отраженной блёсткости, а также неравномерность яркости на рабочем месте. Также следует учесть, что монитор выдает изображения

покадрово с высокой частотой смены кадров, что можно определить как повышенный уровень пульсации светового потока;

б) химические. К химическим факторам можно отнести наличие в воздухе рабочего пространства различных веществ, таких как двуокись углерода, озон, аммиак, фенол, формальдегид и полихлорированных бифенилов;

в) психофизиологические. К ним относятся напряжение различных органов, напряжение внимания, интеллектуальные и эмоциональные нагрузки, монотонность труда, работа с большим объемом информации.

Работоспособность любого человека зависит от факторов окружающей среды, таких как повышенные уровни запыленности воздуха рабочей зоны, пониженная или повышенная влажность воздуха рабочей зоны, пониженная или повышенная подвижность воздуха рабочей зоны, повышенный или пониженный уровень освещенности.

Освещение играет важную роль в работе, в частности при работе с ЭВМ, где есть яркий источник света, монитор. Для создания оптимальной освещённости в помещении люди используют различные системы для создания естественного освещения внутри рабочих помещений.

Нехватка освещения может привести к снижению функционирования зрительного аппарата, а также может влиять на психику человека, определяя его эмоциональное состояние и усталость, вследствие усилий на распознавание нечётких образов.

Важно заметить, что свет влияет на все основные процессы жизнедеятельности, вплоть до иммунной системы и способности сопротивляться неблагоприятным факторам. Причем, сравнительная оценка различных источников света показала, что наибольшее влияние на человека оказывает именно естественное освещение.

Важно, что все аспекты качества освещения влияют на предотвращение несчастных случаев. Например, неравномерное освещение приводит к проблемам с адаптацией, снижая видимость. При низком уровне освещения работники могут чувствовать усталость глаз и переутомление, что является причинами снижения работоспособности. На основе всего вышесказанного можно сделать вывод, что неправильное освещение является значительной угрозой в работе. [32]

Различают естественное, искусственное и совмещённое освещение.

Чаще на предприятиях применяется совмещённое освещение, потому что рабочее время в основном в светлое время суток, когда необязательно заменять естественное освещение искусственным, что в свою очередь ведет к экономии электроэнергии. Однако немаловажным является и создание искусственного освещения на случаи раннего наступления ночи, неблагоприятной погоды и других факторов, когда естественного освещения недостаточно.

5.3 Расчет освещения рабочего места

Естественным освещением считается то, при котором основным источником света выступает Солнце. Искусственное же освещение обеспечивается при помощи искусственных источников освещения – химических, электрических, электролюминесцентных и других.

Для вычисления уровня естественной освещённости в светлое время суток важно учитывать географическое расположение дома и его положения на местности, расположение и направление окон и другие факторы, которые определяются по справочной литературе, а при расчетах принимаются в виде переменных.

5.3.1 Таблицы со значениями

В этом разделе представлены таблицы 5.1-5.10, необходимые для расчетов освещённости.

Таблица 5.1 - Значения коэффициента t_1

Вид светопропускающего материала	t_1
Стекло оконное листовое:	
одинарное	0,9
двойное	0,8
тройное	0,75
Органическое стекло:	
прозрачное	0,9
молочное	0,6
Стеклопакеты	0,8

Таблица 5.2 - Значения коэффициента t_2

Вид переплета для окон промышленных зданий	t_2
Переплеты деревянные:	
одинарные	0,75
спаренные	0,7
двойные раздельные	0,6
Переплеты стальные:	
одинарные открывающиеся	0,75
одинарные глухие	0,9
двойные открывающиеся	0,6
двойные глухие	0,8

Таблица 5.3 - Значения коэффициента t_3

Несущие конструкции покрытий	t_3
Железобетонные и деревянные формы и арки	0,8

Таблица 5.4 - Значение коэффициента t_4

Солнцезащитные устройства, изделия и материалы	t_4
Убирающиеся регулируемые жалюзи и шторы (межстекольные внутренние, наружные)	1

Таблица 5.5 - Значение коэффициента g_1

Отношение глубины помещения к высоте от уровня условной рабочей поверхности верха окна	Отношение расстояния расчетной точки от наружной стены к глубине помещения	Значения g_1 при боковом двустороннем освещении		
		Средневзвешенный коэффициент отражения потолка, стен и пола – 0,5		
		Отношение длины помещения к его глубине		
		0,5	1	2 и более
1	2	12	13	14
От 1 до 1,5	0,1	1,05	1,05	1,05
	0,5	1,35	1,25	1,15
	1	1,6	1,4	1,25
Свыше 1,5 до 2,5	0	1,05	1,05	1,05
	0,3	1,3	1,2	1,1
	0,5	1,8	1,45	1,25
	0,7	2,1	1,75	1,5
	1	2,35	2	1,6

Таблица 5.6 - Значение коэффициента запаса K_3

Помещения и территории	Примеры помещений	Коэффициент запаса K_3				
		при ЕО и расположении светопропуск. материала			При искусственном освещении	
		Вертикально	Наклонно	Горизонтально	Газоразрядные лампы	Лампы накаливания
Помещения общественных и жилых зданий.	Кабинеты, учебные помещения, лаборатории, торговые залы и т.д.	1,2	1,4	1,5	1,5	1,3

Таблица 5.7 - Значения коэффициента $K_{зд}$

$P/H_{зд}$	0,5	1	1,5	2	3 и более
$K_{зд}$	1,7	1,4	1,2	1,1	1

Таблица 5.8 - Значения световой характеристики η_o окон при боковом освещении

Отношение длины помещения к его глубине	Значения световой характеристики η_o при отношении глубины помещения к его высоте от уровня условной рабочей поверхности до верха окна							
	1	1,5	2	3	4	5	7,5	10
4 и более	6	7	7,5	8	9	10	11	12,5
3	7,5	8	8,5	9,6	10	11	12,5	14
2	8,5	9	9,5	10,5	11,5	13	15	17
1,5	9,5	10,5	13	15	17	19	21	23
1	11	15	16	18	21	23	26,5	29
0,5	18	23	31	37	45	54	66	-

Таблица 5.9 - Нормы освещённости e_n

Разряд зрит. Работы	Подраз зрит. работы	Контракт объекта с фоном	Характе- ристика фона	Искусственн ое освещение	Естественное освещение
				Освещенность, лк	КЕО, e_n , %
				при системе общего освещ.	при боков. освещ.
III	a	Малый	Темный	300	1.5

Таблица 4.10 - Коэффициент светового климата, m

Световые проемы	Ориентация световых проемов по сторонам горизонта	Коэффициент светового климата, m
		Алматы
В наружных стенах зданий	С	0,75
	СВ, СЗ	0,7
	З, В	0,65
	ЮВ, ЮЗ	0,65
	Ю	0,65

5.3.2 Расчёт естественного освещения

Для примера был взят кабинет жилого дома расположенный в углу дома, выходящий внешними стенами на север и восток. На рисунке 5.2 представлен план этого помещения с размерами, указанными в миллиметрах, для которого необходимо провести расчет уровня естественной освещенности.

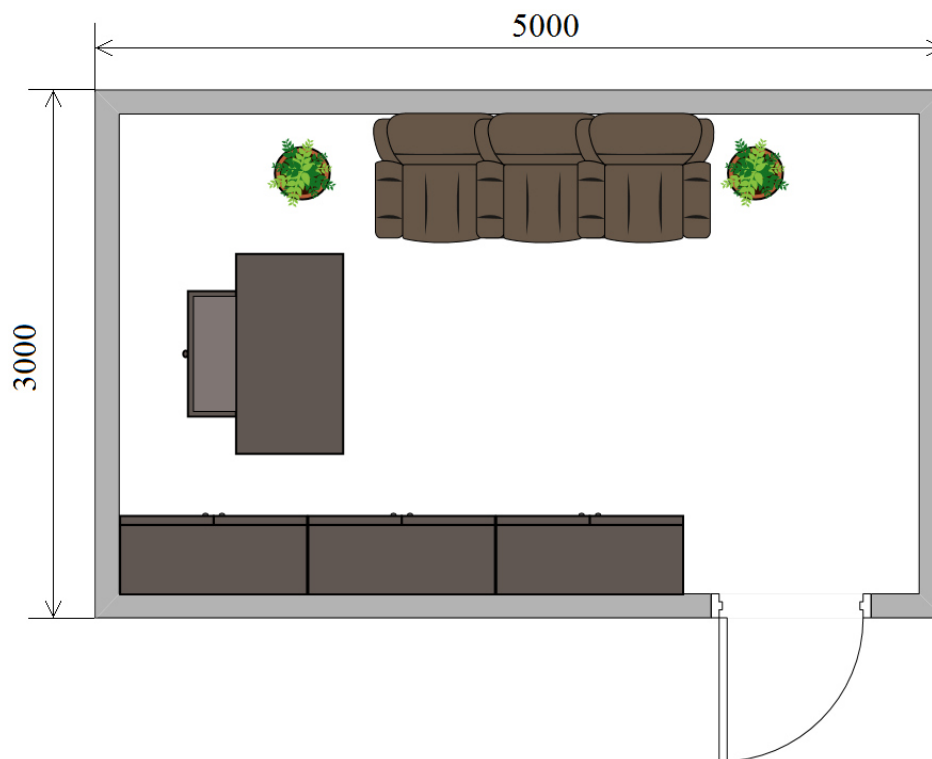


Рисунок 5.2 – План помещения

Как следует из плана помещения, длина помещения составляет 5 м, глубина – 3 м, высота – 2.7 м. Высота условной рабочей поверхности (стола) составляет 0.75 м, высота от пола до окна составляет 0.9 м, высота оконных проемов – 1200 мм, ширина оконных проемов – 900 мм, высота окон – 1170 мм, ширина окон – 870 мм. Окна – двустворчатые, с деревянными переплетами.

Освещенность измеряется в люксах (лк). Естественная освещенность в помещении имеет такой важный параметр, как коэффициент естественной освещенности (КЕО) e , измеряющимся в процентах [34].

$$e = \frac{E_B}{E_H} \cdot 100\%, \quad (5.1)$$

где: E_B – одновременная освещенность рассеянным светом снаружи, лк;
 E_H – освещенность внутри помещения, лк.

Уровень естественной освещенности при боковом освещении (так как в помещении имеются только лишь окна рассчитывается по формуле (5.2) [35]:

$$100 \cdot \frac{S_o}{S_n} = \frac{e_N K_3 \eta_o}{\tau_o r_1} \cdot K_{зд}, \quad (5.2)$$

где: S_o – площадь световых проемов при боковом освещении, м²;
 S_n – площадь пола помещения, м²;
 e_N – нормируемое значение КЕО;
 K_3 – коэффициент запаса, принимают по таблице 5.6;
 η_o – световая характеристика окон, принимают по таблице 5.8;
 r_1 – коэффициент, учитывающий повышение КЕО при боковом освещении, благодаря свету, отраженному от поверхности помещения и подстилающего слоя, примыкающего к зданию, принимают по таблице 5.5;
 $K_{зд}$ – коэффициент, учитывающий затемнение окон противостоящими зданиями, принимают по таблице 5.7;
 τ_o – общий коэффициент светопропускания.

Общий коэффициент светопропускания, в свою очередь, определяют по формуле(5.3):

$$\tau_o = \tau_1 \tau_2 \tau_3 \tau_4 \tau_5, \quad (5.3)$$

где: τ_1 – коэффициент светопропускания материала, по таблице 5.1;
 τ_2 – коэффициент, учитывающий потери света в переплетах светопроема, принимают по таблице 5.2;
 τ_3 – коэффициент, учитывающий потери света в несущих конструкциях, при боковом освещении равен 1, при верхнем освещении принимают по таблице 5.3;
 τ_4 – коэффициент, учитывающий потери света в солнцезащитных устройствах, принимают по таблице 5.4;
 τ_5 – коэффициент, учитывающий потери света в защитной сетке, устанавливаемой под фонарями, принимают равным 0,9.

Нормированное значение КЕО с учетом расположения здания в определенном географическом регионе с особенностями обеспечения естественным светом рассчитывается согласно формуле (5.4):

$$e_N = e_n \cdot m_N, \quad (5.4)$$

где: e_n – значение КЕО согласно таблице 5.9;
 m_N – коэффициент светового климата согласно таблице 5.10.

Коэффициент светового климата учитывает интенсивность солнечной радиации в разных географических регионах и направленность световых проемов в те или иные стороны света. Например, южные регионы Казахстана, естественно, получают гораздо больше солнечного света, а если световые проемы в комнате выходят на юг, то в световой проем будет попадать больше света в течение дня.

Для расчета уровня естественной освещенности сперва вычислим нормированное значение КЕО:

$$e_N = 1.5 \cdot 0.75 = 1.125.$$

Для расчета общего коэффициента светопропускания, необходим подбор значений составляющих его коэффициентов. Примем окно как окно со стеклопакетом, с одинарным деревянным переплетом, со сплошной рамой, убирающимися жалюзи.

$$\tau_o = 0.8 \cdot 0.75 \cdot 0.8 \cdot 1 \cdot 0.9 = 0.432.$$

Площадь комнаты получим перемножив длину на ширину, то есть 15 м^2 . Так как плотность застройки в частном жилом секторе низкая коэффициент затенения берётся за 1, согласно таблице. Офис находится на втором этаже, отношение высоты карниза соседнего здания по сравнению с высотой окон кабинета и расстояния между соседними домами с малой вероятностью будет менее 3. Коэффициент запаса жилого помещения, согласно таблице принимается равным 1.5. Значение коэффициента световой характеристики η_o принимается согласно таблице исходя из двух отношений: отношения длины комнаты к его глубине, и отношения глубины помещения к его высоте от высоты условной рабочей поверхности до верха окна (1.32 м). Первое отношение равно 1.67, которое можно округлить до 2, второе – 2.27, которое также следует округлить то стандартного значения 2. По полученным отношениям подбираем значение 9.5. Для определения коэффициента r_1 необходимо знать несколько параметров. Необходимо определить: отношение длины помещения к его глубине; отношение глубины помещения к его высоте от условной рабочей поверхности до верха окна, которое будет равно 2.27; отношение расстояния от наружной стены до расчетной точки (1.5 м) к глубине помещения (3 м), которое будет равно 0.5. Также необходимо знать коэффициент отражения стен и потолка, который будет равен 0.5 у белых стен и потолков. Также учитывается то, что освещение является двусторонним, так как кабинет является угловым помещением в доме и внешними стенами выходит на север и на восток, и следовательно, существует возможность расположения оконных проемов во внешних стенах с двух сторон.

Исходя из этих параметров, по таблице принимаем в качестве значения коэффициента значение 1.25.

Имея все нужные коэффициенты, можно приступить к расчету уровня естественной освещенности. Для этого нужно преобразовать формулу (5.2) с учетом того, что необходимо вычислить именно минимальную необходимую площадь окон.

$$S_o = \frac{S_n e_N K_3 \eta_o}{100 \tau_o r_1} K_{зд} = \frac{15 \cdot 1.125 \cdot 1.5 \cdot 9.5}{100 \cdot 0.432 \cdot 1.25} \cdot 1 = 4.45 \text{ м}^2.$$

Количество необходимых в комнате световых проёмов вычисляется по формуле:

$$N = \frac{S_o}{S_x}, \quad (4.5)$$

где: S_x – площадь одного окна.

Площадь отдельного окна будет равна произведению его высоты и ширины:

$$S_x = 1.17 \cdot 0.85 = 0.994.$$

Тогда необходимое количество окон будет равно:

$$N = \frac{4.45}{0.994} = 4.48 \approx 5.$$

Исходя из этого, план помещения с окнами представлен на рисунке 5.3.

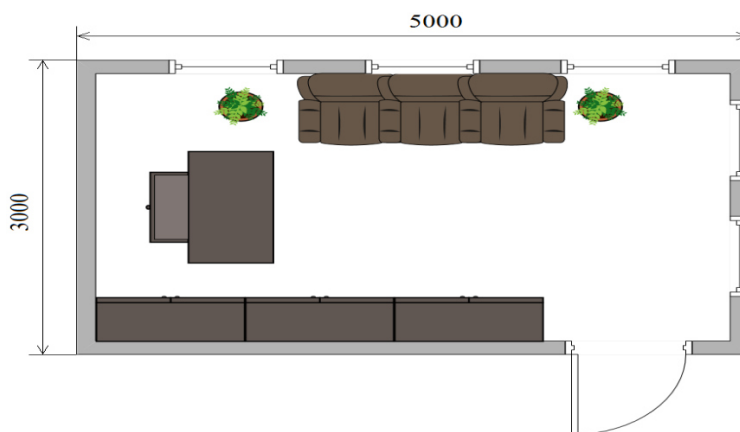


Рисунок 5.3 – План помещения с окнами

Три окна были добавлены с северной стороны, и два окна с восточной. Это позволит обеспечить равномерное и оптимальное освещение в рабочем кабинете на достаточном уровне, чтобы в помещении можно было выполнять письменные работы.

5.3.3 Расчет искусственного освещения рабочего помещения

Как уже говорилось искусственное освещение обеспечивается при помощи искусственных источников освещения.

Уровень искусственного освещения в помещениях возможно рассчитать при помощи нескольких разных методов: расчет методом коэффициента использования светового потока, расчет по удельной мощности, либо метод точечного расчета.

Метод точечного расчета является весьма трудоемким, так как рассчитывает уровень освещенности в какой-либо точке путем расчета суммы световых сил, излучаемых в эту точку от каждого светильника, освещающего её. А метод расчета по удельной мощности в основном применяется тогда, когда необходимо предварительно рассчитать мощность осветительного оборудования, когда необходимо проверить точность светотехнических расчетов, либо когда проектируется освещение помещений, не предназначенных для точных работ или комфортного обитания. Поэтому во время расчетов, с учетом большего удобства использования, будет применяться метод расчета по световому потоку. Также этот метод хорошо подходит для данной задачи потому, что он учитывает также и световой поток, отраженный от различных поверхностей, и весьма эффективен при подсчете освещенности в помещениях со светлыми стенами и потолками. Метод светового потока позволяет подсчитать общее количество светильников, требуемых для того, чтобы создать нужное количество светового потока на единицу площади в помещении, при условии равномерного их распределения.

На рисунке 5.2 уже был представлен план помещения, для которого рассчитывался уровень естественной освещенности, и для него же необходимо провести расчет количества светильников, при котором будет обеспечиваться требуемый световой поток на всей площади помещения.

В роли светильника был выбран двухламповый люминесцентный светильник ЛПО46-2х36-712. Его характеристики приведены в таблице 5.11.

Таблица 5.11 – Характеристики люминесцентной лампы ЛБ-40

Тип лампы	Мощность, Вт	Напряжение, В	Ток лампы, А	Световой поток, лм	Длина лампы, мм	Диаметр лампы, мм
ЛБ-40	40	103	0.43	3000	1213	40

Необходимое количество светильников для помещения рассчитывается по формуле (5.6):

$$N = \frac{E_{min} \cdot S \cdot z \cdot K}{\Phi \cdot \eta}, \quad (5.6)$$

где: Φ – расчетный световой поток светильника, лм;

E_{min} – минимальная необходимая освещенность поверхности в помещении, выбирается согласно таблице норм освещенности 5.9, лк;

S – площадь помещения, м²;

z – коэффициент минимальной освещенности (отношение средней и минимальной освещенности), для светодиодных светильников он будет равен 1;

K – коэффициент запаса;

η – коэффициент использования светового потока, который для люминесцентных ламп берется равным 26.

Минимальная необходимая освещенность выбирается согласно разряду и подразряду зрительных работ. Так как подразумевается, что кабинет используется в основном для чтения и письменных работ, в особых случаях – для написания чертежей, разряд и подразряд зрительных работ будут взяты как Коэффициент η выбирают по типу светильников и значению индекса помещения i , который отражает влияние на коэффициент использования размеров помещения:

$$i = \frac{S}{h \cdot (A + B)}, \quad (5.7)$$

где h - высота светильников над рабочей поверхностью;

A, B, S – соответственно, длина, ширина и площадь помещения.

Тогда индекс помещения будет равен:

$$i = \frac{15}{1.95 \cdot (5 + 3)} = 0.96.$$

По значению индекса помещения, а также по типу светильников и коэффициентам отражения, для коэффициента использования светового потока выбирается значение 26.

Получив все нужные коэффициенты, можно приступить к расчету количества светильников:

$$N = \frac{500 \cdot 15 \cdot 1 \cdot 1.5}{200 \cdot 26} = 2.163 \approx 2.$$

Таким образом, для освещения комнаты достаточно лишь двух светодиодных ламп.

Схема расположения светильников будет выглядеть так, как представлено на рисунке 5.4.

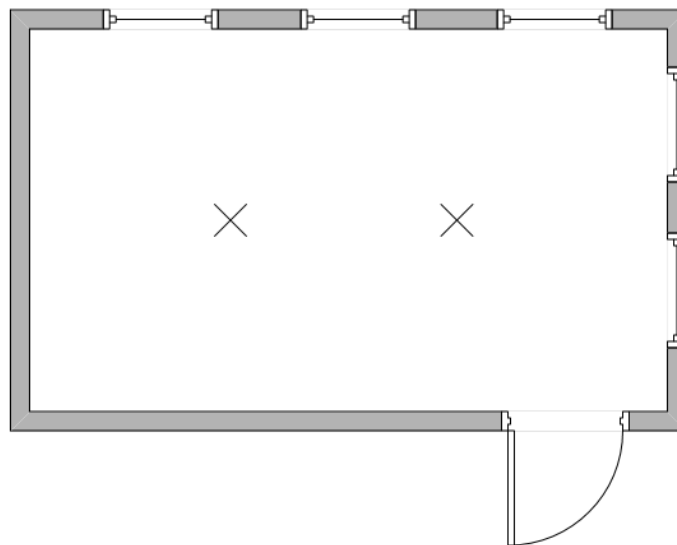


Рисунок 5.4 – Размещение светильников в помещении

Заключение

В ходе проведённой работы был проведён обзор теории распознавания образов и идентификации людей. Согласно данным наиболее удобным и быстрым в реализации и применении является метод Виолы-Джонса, на основе которого и была выполнена данная дипломная работа. Данный метод базируется на технологии нейронных сетей, которая значительно упрощает задачу детектирования объектов на изображении.

В соответствии поставленной задаче разработки программного продукта был разработан собственный алгоритм распознавания личности, процесс создания которого и дальнейшая проверка которого представлены в третьей главе. При использовании открытого ресурса библиотеки OpenCV программный продукт работает успешно.

Согласно вышесказанному можно сделать вывод, что все поставленные в начале работы задачи были выполнены успешно. Полученные результаты показывают, что технология искусственных нейронных сетей позволяет быстро решить задачу идентификации, что может быть использовано практически повсеместно в жизни человека. Однако следует отметить, что в данной работе были использованы нейронные сети простейшей архитектуры, и при усложнении архитектуры нейронных сетей, применении более широких моделей, применении дополнительных методов возможно добиться большей точности при распознавании образов на изображении в целом. Кроме того развитие нейронных сетей приводит к снижению требований к аппаратным возможностям ЭВМ.

При применении полученного продукта на практике возможно вынесение базы данных лиц и алгоритмы сложных математических вычислений на общий сервер, обладающий большей производительностью, чем персональный компьютер, что в свою очередь повысит эффективность работы нейронной сети.

В качестве дальнейшего развития данной работы может быть предложено распознавание предположительного возраста и пола человека, создание усреднённого лица по характеристикам выбранных людей, применение маски на лицо и замена лиц, в том числе и в режиме реального времени. Для этого возможно использовать иные архитектуры нейронных сетей, такие как, например, рекуррентные нейронные сети, показывающие большую точность, в сравнении со свёрточными ИНС.

Список литературы

- 1 Информационный портал Life.ru – https://life.ru/t/технологии/1099625/s_pomoshchiu_ii_kitaitsy_poimali_priestupnika_vsiegho_za_dvie_minuty (дата обращения: 20.03.2019).
- 2 Г.А. Кухарев, Н.Л. Щеголева., Системы распознавания человека по изображению лица /– СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2006. – 176 с.
- 3 Ю.В. Визильтер, С.Ю. Желтов, А.В. Бондаренко и др., Обработка и анализ изображений в задачах машинного зрения – М.: Физматкнига, 2010. – 672 с.
- 4 Van Veen F. The neural network zoo //The Asimov Institute, 2016.
- 5 McCulloch W. S., Pitts W. A logical calculus of the ideas immanent in nervous activity //The bulletin of mathematical biophysics. – 1943. – Т. 5. – №. 4. – С. 115–133.
- 6 Ripley B. D. Pattern recognition and neural networks. – Cambridge university press, 2007.
- 7 LeCun Y. et al. Gradient-based learning applied to document recognition //Proceedings of the IEEE. – 1998. – Т. 86. – №. 11. – С. 2278–2324.
- 8 Фрэнк Розенблатт. Принципы нейродинамики: перцептроны и теория механизмов мозга = Principles of Neurodynamic: perceptrons and the theory of brain mechanisms. — М.: «Мир», 1965.
- 9 Твердотельная революция в телевидении. Телевизионные системы на основе приборов с зарядовой связью, систем на кристалле и видеосистем на кристалле. Березин В.В., Умбиталиев А.А., Фахми Ш.С., Цыцулин А.К., Шипилов Н.Н.; под ред. Умбиталиева А.А. и Цыцулина А.К. М.: Радио и связь, 2006. 312 с.
- 10 Горбачёв, А. А. Твердотельные матричные фотопреобразователи и камеры на их основе: учебное пособие/А. А. Горбачёв, В. В. Коротаев, С. Н. Ярышев – СПб.: НИУ ИТМО, 2013. – 98 с.
- 11 Ту Дж., Гонсалес Р. Принципы распознавания образов // Пер. с англ. Гуревич И. Б. – М.: «Мир», 1978.
- 12 Маткасым Н. Н. Распознавание образов с помощью нейронных сетей //Технологии Microsoft в теории и практике программирования: сборник трудов XIII Всероссийской научно-практической конференции студентов, аспирантов и молодых ученых, г. Томск, 22–23 марта 2016 г.—Томск, 2016. – С. 23–25;
- 13 Чабан Л. Н. Теория и алгоритмы распознавания образов. Учебное пособие. – М.: МИИГАиК, 2004, 70 с.
- 14 Viola P., Jones M. Rapid object detection using a boosted cascade of simple features //Computer Vision and Pattern Recognition, 2001. CVPR 2001. Pro-

ceedings of the 2001 IEEE Computer Society Conference on. – IEEE, 2001. – Т. 1. – С. 137–138.

15 Е.С. Мищенко. Сравнительный анализ алгоритмов распознавания лиц – Труды молодых учёных, Волгоградский государственный университет, Волгоград, Российская Федерация, 2013.

16 Viola P., Jones M. J. Robust real-time face detection //International journal of computer vision. – 2004. – Т. 57. – №. 2. – С. 137–154.

17 Самоучитель Python: <https://pythonworld.ru/samouchitel-python> (дата обращения: 1.01.2019 - 1.05.2019).

18 Сайт Intel: <https://www.intel.ru> (дата обращения: 2.02.2019).

19 Документация библиотеки OpenCV : <https://docs.opencv.org/> (дата обращения: 20.01.2019).

20 Gary Bradski, Adrian Kaehler. Learning OpenCV. Computer vision with the OpenCV library. 2008.

21 Официальный сайт Алматинского университета энергетики и связи: <https://aues.kz> (дата обращения: 20.04.2019).

22 Бекишева А.И. Методические указания к выполнению экономической части дипломной работы бакалавров специальности 5В0703 – Информационные системы – Алматы: АУЭС; 2013. -24с.

23 Параграф-WWW. Производственный календарь на 2019 год - https://online.zakon.kz/Document/?doc_id=34186053#pos=65;-51 (дата обращения: 20.03.2019).

24 HeadHunter - hh.kz (дата обращения: 20.03.2019).

25 Анализ миграционной подвижности населения Республики Казахстан по итогам национальной переписи населения 2009 года – Центр исследований Сандж, Астана, 2011.

26 Аналитический доклад на тему: «Актуальные вопросы миграционной политики: оценка текущего состояния, прогноз потребности в трудовых мигрантах с учетом их квалификации, нелегальная трудовая миграция, опыт зарубежных стран. Министерство национальной экономики Республики Казахстан. Астана, 2014.

27 Родионова Н.А. Нелегальная миграция как социально-экономическая проблема российского общества. Челябинский гуманитарий, 2011, №2(15).

28 Неклюдова Н.П., Илибаева Е.А. Оценка потерь бюджета региона от нелегальной трудовой миграции (на примере свердловской области). Вестник Томского государственного университета. Экономика. 2017. №37.

29 Указ Президента Республики Казахстан "О Концепции миграционной политики Республики Казахстан на 2007-2015 годы", Астана, 2007.

- 30 Закон Республики Казахстан от 30 ноября 2018 года № 197-VI «О республиканском бюджете на 2019-2021 годы».
- 31 Налоговый Кодекс РК 25 декабря 2017 года № 120-VI ЗРК.
- 32 Приходько Н.Г. Основы безопасности жизнедеятельности. Конспект лекций для студентов всех специальностей. – Алматы, АУЭС, 2011. – 40 с.
- 33 Абрамов В.В. Безопасность жизнедеятельности. Учебное пособие для вузов. — Санкт-Петербург: 2013. — 365 с.
- 34 СНиП РК 2.04-05-2002. Естественное и искусственное освещение. Общие требования. Комитет по делам строительства Министерства индустрии РК - Астана, 2002г.
- 35 Абдимуратов Ж.С., Мананбаева С.Е. Безопасность жизнедеятельности. Методические указания к выполнению раздела «Расчет производственного освещения» в выпускных работах для всех специальностей. Бакалавриат – Алматы: АИЭС, 2009. – 20 с.

Приложение Б

В приложении А представлен код программы, идентифицирующей лица.

```
import cv2
import numpy as np
import os

recognizer = cv2.face.LBPHFaceRecognizer_create()
recognizer.read('trainer/trainer.yml')
cascadePath = "haarcascade_frontalface_default.xml"
faceCascade = cv2.CascadeClassifier(cascadePath);

font = cv2.FONT_HERSHEY_SIMPLEX

id = 0

names = ['None', 'It is Me', 'Balbaev', 'Auesova', 'Adambaev', 'Djeksenbinov',
        'Baikenov', 'Golubeva']

cam = cv2.VideoCapture(0)
cam.set(3, 640)
cam.set(4, 480)
minW = 0.1*cam.get(3)
minH = 0.1*cam.get(4)

while True:
    ret, img =cam.read()
    gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)

    faces = faceCascade.detectMultiScale(
        gray,
        scaleFactor = 1.2,
        minNeighbors = 5,
        minSize = (int(minW), int(minH)),
    )
    for(x,y,w,h) in faces:
        cv2.rectangle(img, (x,y), (x+w,y+h), (0,255,0), 2)
        id, confidence = recognizer.predict(gray[y:y+h,x:x+w])
```

Продолжение приложения Б

```
if (confidence < 100):
    id = names[id]
    confidence = " {0}%".format(round(100 - confidence))
else:
    id = "unknown"
    confidence = " {0}%".format(round(100 - confidence))

cv2.putText(img, str(id), (x+5,y-5), font, 1, (255,255,255), 2)
cv2.putText(img, str(confidence), (x+5,y+h-5), font, 1, (255,255,0), 1)

cv2.imshow('camera',img)

k = cv2.waitKey(10) & 0xff
if k == 27:
    break

print("\n [INFO] Exiting Program and cleanup stuff")
cam.release()
cv2.destroyAllWindows()
```