Автор: **Останин Иван Васильевич**

студенты гр. ПКС-306 ГАПОУ СО "Ирбитский политехникум"

Специальность: "09.02.03 Программирование в компьютерных системах"

Руководитель: преподаватель ИПТ Вишнякова Наталья Викторовна

Тема проекта: «**Термометрия, оптическая передача данных с термометра на языке Python**»

Срок реализации проекта: с 1 октября 2021 по 31 марта 2022 года.

В нашей жизни нередко возникает потребность фиксации показаний различных приборов в определенные промежутки времени. Это может быть измерение давления, веса, сил светового потока, силы тяжести, а также температуры.

Раздел термометрии в физике позволяет разрабатывать методы и средства измерения температуры.

При проведении различных химический и физических экспериментов необходимо протоколировать температуру процесса на определённом промежутке времени. Снятие показаний температуры должно выполняться точно и с определенной частотой. Вручную данный процесс производить неэффективно и нецелесообразно.

Поэтому возникла необходимость автоматизации процесса, а именно создания программного продукта, позволяющего в автоматическом режиме передавать показания с дисплея термометра и сохранения этих показаний в различных форматах представления - в виде текстовый форматов обмена данными, таблицы, графика и д.п.

Для реализации поставленной задачи был выбран язык программирования Python - один из самых востребованных в мире.

Python обладает огромной базой пользовательских библиотек и расширений, созданных сообществом и находящихся в открытом доступе, что позволяет использовать их в программах и увеличивает скорость разработки. Работа с сайтами, различные боты для социальных сетей, построение сложнейших приложений и кроссплатформенная разработка. Стоит отметить и то, что он обладает крайне простым синтаксисом и очень прост в освоении.

**Цель проекта** - создание программного средства для оптической передачи данных с термометра на языке Python.

Для выполнения цели поставлены следующие **задачи**:

* подобрать и систематизировать литературу по теме;
* изучить принципы формирования изображения термометра TP300;
* рассмотреть особенности разработки программного продукта на языке Python;
* разработать интерфейс взаимодействия пользователя с программой;
* разработать алгоритм, реализовать его на языке программирования Python и провести тестирование и отладку программного средства;
* оформить руководство оператора на программный продукт.

Объектом исследования является цифровой термометр. Предметом, язык программирования Python.

**Этапы реализации проекта**

**1. Постановка задачи**

Оптическая передача данных с термометра предполагает следующие два этапа: 1) фиксация показаний термометра на видео, 2) обработка видеофайла. Программный средство должно выполнить последний шаг - обработать видеофайл, произвести перевод графических данных с кадра изображения в числовой ряд и выполнить экспорт данных в выбранный пользователем формат.

Для этого программа реализует задачи:

1. Получить из видеофайла данные о состоянии каждого сегмента дисплея
2. Дешифровать данные в числовой вид
3. Экспортировать дешифрованные данные.

Для решения первой задачи использовался метод ключевых точек изображения. На основании цвета пикселя формируются данные о состоянии сегментов.

Для решения второй была создана таблица, содержащая состояния всех сегментов внутри цифры на дисплее. Для дешифрации набора состояний сегментов в цифру, будет достаточно сопоставить его со строками в таблице.

**2. Разработка приложения**

**2.1 Разработка структуры программы**

Программа разрабатывалась в методологии объектно-ориентированного программирования. Благодаря этому программу можно разделить на несколько независимых частей, каждая из которых играет определённую роль. Были созданы пять классов, между которых распределили функции программы:

VideoScanner - основной класс, который отвечает за обработку графической информации

App - класс приложения отвечающий за его настройку и экспорт выходных данных

Digit, Segment - предоставляют доступ к методам для работы с цифрами и сегментами на экране термометра

Interupt – класс дешифровки данных с дисплея в числа

Работа программы основана на расстановке ключевых точек на видео над каждым сегментом дисплея термометра. На основании цвета пикселя на кадре под сегментами программа распознает состояние каждого сегмента на экране дисплея и как следствие - число.

Работа с программной разделена на пять этапов:

* Transforming – предварительная подготовка видео, происходит настройка масштабирования и вращения кадра видео;
* Placement – расстановка пользователем ключевых точек;
* Naming – присваивание каждой точке имени, положения её сегмента в семисегментном индикаторе и распределения точек между различными цифрами дисплея. Присваивание положения сегмента внутри цифры идёт по порядку сверху вниз и слева направо. То есть сначала пользователь нажимает на верхний сегмент, за ним на левый верхний, правый верхний, средний, нижний левый, нижний правый, нижний. Наименование цифр идёт по порядку слева направо. Названные сегменты приобретают зелёную обводку, а сегменты, входящие в заполненную цифру, синюю (Рис.1).

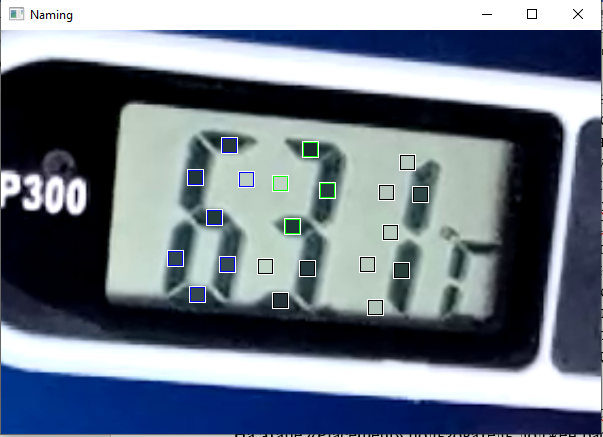


Рис. 1 Пример названных ключевых точек.

* Scanning – этап сканирования в котором на основании всех ключевых точек собирается информация о состоянии каждого элемента индикатора, эта информация дешифруется в формат целого числа и экспортируется в удобном формате. Сегменты, входящие в цифру, которую удалось интерпретировать без проблем, имеют синюю обводку, а сегменты чья интерпретация завершилась с ошибкой жёлтую (Рис. 2);



Рис. 2 Пример сканирования с ошибкой.

* Fixing – коррекция положения ключевых точек, на случай если в процессе исследования температуры камера и термометр сместились друг относительно друга и расположение сегментов на изображении изменилось. (Рис. 3)

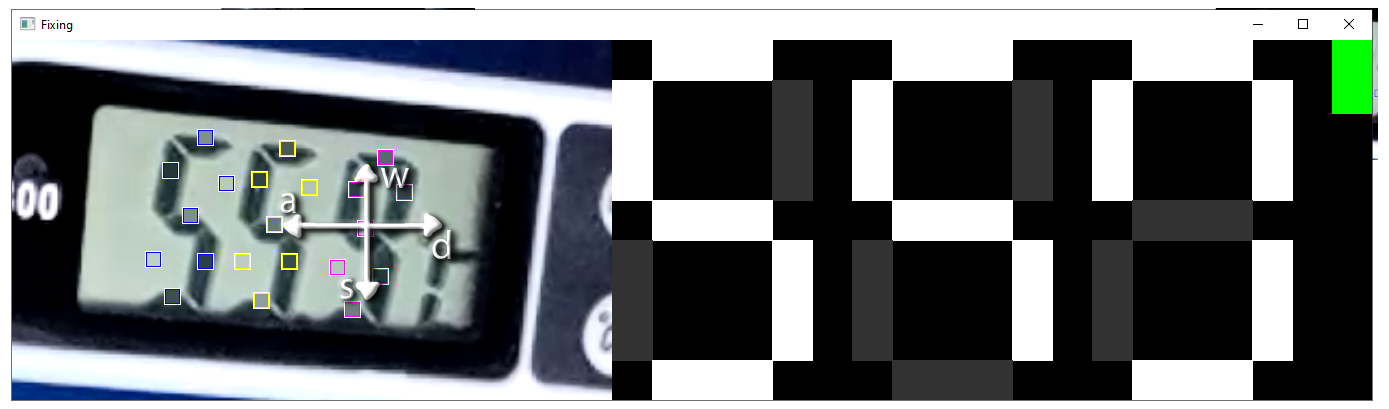


Рис. 3 Пример коррекции ошибки.

Во время разработки приложения возникла проблема чёткого поиска. Так как дешифратор ищет точное совпадение полученных данных с одной из строк в таблице, то при ошибке измерения состояния хотя бы одного сегмента, выходное значение будет непригодно для использования.

Решением проблемы - реализация алгоритма нечёткого поиска. Существует множество его алгоритмов, в данном случае был реализован следующий: если чёткого совпадения не было найдено создаётся список нулей длинной 10 элементов. И для каждой строки в таблице dataSet вычисляется значение какого количества сегментов не совпало с дешифруемыми значениями. Полученные вычисления сохраняются в список. После чего в данном списке находится минимальное значение, индекс которого и будет результатом нечёткого поиска.

**2.2 Тестирование прилдожения**

Следующим этапом стало Тестирование. Тестирование проводилось на трёх экспериментах:

E-Test – чёрно-белое видео созданное с помощью компьютерной графики;

E-ColorTest – цветное видео являющиеся копией видео эксперимента E-Test за исключением того что цвета для фона и сегментов максимально приближены к реальным;

E-1 – первый реальный эксперимент.

На данном слайде вы можете видеть результаты экспериментов. Во время проведения Е-1 Внешние обстоятельства внесли в исходное видео такие дефекты как: расфокус камеры, авто-выключение термометра на 550 секунде и изменение уровня освещения. Из-за всего вышеперечисленного удачное сканирование не было возможно изначально. Но в целом, считаю, результат удовлетворительным.

Результатом курсовой работы является разработанная программа для дешифровки данных с дисплеев электронных устройств. Данный продукт позволяет: экспортировать данные в различные форматы, проводить коррекцию ошибок на этапе сканирования. Одним из достоинств программы является простота в освоении и не требование от пользователя специальных знаний.

В процессе выполнения курсовой работы закреплены навыки объектно-ориентированного программирования, получен практический опыт разработки алгоритма, поставленной задачи средствами языка программирования Python, выполнена отладка и тестирование программы на уровне модуля.

Спасибо за внимание. Я готов ответить на ваши вопросы.