

Министерство науки и высшего образования Российской Федерации
ФГБОУ ВО АЛТАЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Институт цифровых технологий, электроники и физики
Кафедра вычислительной техники и электроники (ВТиЭ)

Отчёт по:

ПРОЕКТНО-ТЕХНОЛОГИЧЕСКОЙ ПРАКТИКЕ

Студент группы: _____ 505 _____ Э. Н. Плотицына

Руководитель работы: _____ ст. препод. _____ И. А. Шмаков

Барнаул 2021 г.

РЕФЕРАТ

Полный объем работы составляет 19 страниц, включая 0 рисунков и 0 таблиц.

Цель работы (формулировка задания):

Создать приложение для считывания текста с изображения и занесения его в таблицу. Программа нацелена на работу с изображениями гербарных образцов, имеющих этикетку с сопровождающей информацией. Распределение информации об образце должно быть максимально простым в использовании. В приложении должны присутствовать как заполнение, так и изменение уже существующих табличных данных. Программа должна являться полностью завершенным программным продуктом.

Постановка задачи:

Написать приложение удобное в использовании. На вход программы поступает изображение или .csw файл. Программа считывает текст с изображения и выводит его в виде списка. У пользователя приложение есть возможность редактировать и распределить данных по категориям. При сохранении на выход программы поступает .csv файл, подготовленный к последующей загрузки в электронную базу данных «Virtual Herbarium ALTB». Цель программы помочь ускорить процесс заполнения электронной библиотеки. Время затраченное на анализ изображения программой зависит от характеристик компьютера.

Функциональность и интерфейс пользователя для разрабатываемого программного продукта:

Пользовательский интерфейс представляет собой совокупность программных и аппаратных средств, обеспечивающих диалог пользователя и компьютера.

Для пользователя доступно шесть кнопок, пять из которых активны с запуска приложения. Кнопка "Сканировать изображение" появляется после открытия через файловый диалог изображения.

Функции кнопок:

"Открыть изображение" - выбрать путь расположения файла и открыть его через файловый диалог(для изображения);

"Открыть .csv файл" - выбрать путь расположения файла и открыть его через файловый диалог(для .csv);

”Взять выбранные элементы” - выбрать выделенные строки из списка и переместить данные в поле ”Данные” для редактирования;

”Обновить” - обновит содержимое столбца ”Данные” изменяемой строки таблицы;

”Сканировать изображение” - запуск считывания текста с изображения;

”Сохранить” - сохранить информацию из таблицы в .csv файл.

Последовательность действий для пользователя:

При редактировании .csv файла без необходимости сканирования изображения:

- 1) Открыть файл с табличными данными в формате .csv;
- 2) Выбрать двойным щелчком мыши из таблицы строку с категорией, которую необходимо изменить;
- 3) В поле ”Данные” отредактировать или добавить информацию об образце;
- 4) Использовать кнопку ”Обновить” для переноса текста в таблицу с данными;
- 5) Повторить при необходимости п.2-4;
- 6) Использовать кнопку ”Сохранить”, выбрать имя и расположения нового файла.

При сканировании изображения:

- 1) Открыть изображение в формате .jpg/.png;
- 2) Использовать кнопку ”Сканировать изображение”, и подождать когда процесс закончится;
- 3) Выбрать двойным щелчком мыши из таблицы строку с категорией, которую необходимо изменить;
- 4) Выбрать один или несколько элементов из списка со считанной информацией;
- 5) Использовать кнопку ”Взять выбранные элементы”, после чего они помещаются в поле для редактирования ”Данные”;
- 6) В поле ”Данные” отредактировать или добавить информацию об образце;
- 7) Использовать кнопку ”Обновить” для переноса текста в таблицу с данными;
- 8) Повторить при необходимости п.3-7;

9) Использовать кнопку "Сохранить", выбрать имя и расположения нового файла.

Пользовательский интерфейс Tkinter состоит из отдельных виджетов. Каждый виджет представлен как объект Python, созданный из таких классов, как `ttk.Frame`, `ttk.Label` и `ttk.Button`.

У виджетов есть параметры конфигурации, которые изменяют их внешний вид и поведение, например текст, отображаемый в поле для ввода.

Tkinter реагирует на ввод пользователя, изменения и нажатие кнопок в программе только при активном запуске цикла событий. Программа запускает цикл обработки событий при ее открытии. Обновление пользовательского интерфейса происходит при обратной связи от пользователя.

`Frame` - это виджет рамки. Его основное предназначение - действовать как разделитель или контейнер для сложных окон. Не предназначен для взаимодействия.

`Label` - это виджет, который отображает текстовую строку.

`Button` - это виджет кнопка. Когда пользователь нажимает кнопку (нажатие левой кнопкой мыши), то вызывается команда, указанная в параметре `-command`.

`Entry` - это виджет, который отображает однострочную текстовую строку и позволяет ее редактировать. Для удобства используют стандартный механизм `-xscrollcommand` для взаимодействия с полосами прокрутки.

`Listbox` - это виджет, который отображает список строк, по одной в каждой строке. При первом создании новый список не имеет элементов. Элементы могут быть добавлены или удалены с помощью команд виджета.

`Treewiew` - это виджет, с помощью которого можно выводить информацию в иерархической или форме таблицы.

`Grid` и `pack` - диспетчеры геометрии, контролирующие, где в пользовательском интерфейсе размещаются виджеты при создании.

Рамки с виджетами (`frame`) расположены с помощью `grid` в сетке размером [2,6].

(1,1) - `Treewiew`, таблица "Категория" - "Данные";

(3,1) - `Listbox`, область для вывода считанной с изображения информации в виде списка;

(5,1) - Entry, поля ввода и редактирования текста, перед занесением в таблицу Treeview;

(1-4,0) - Label, заполненный изображением;

(0,0), (5,0), (2,1), (4,1) - области интерфейса содержащие управляющие кнопки.

Ключевые слова: .

Отчёт оформлена с помощью системы компьютерной вёрстки $\text{T}_\text{E}\text{X}$ и его расширения $\text{X}_\text{E}\text{L}_\text{A}\text{T}_\text{E}\text{X}$ из дистрибутива *TeX Live*.

СОДЕРЖАНИЕ

Введение	8
1 Раздел 1. Общие сведения	9
1.1 Обозначение и наименование программы	9
1.2 Программное обеспечение, необходимое для функционирования программы	9
1.3 Язык программирования, на котором написана программа и используемые модули	9
2 Раздел 2. Функциональное назначение	10
2.1 Классы решаемых задач	10
2.2 Назначение программы	10
3 Раздел 3. Описание логической структуры	11
3.1 Используемые методы	11
3.1.1 Tkinter	11
3.1.2 OpenCV	12
3.1.3 Pillow	12
3.1.4 Csv	12
3.1.5 EasyOCR	13
3.2 Описание функций составных частей программы	13
3.2.1 Функции главной программы	13
3.2.2 Функции подмодуля программы	13
3.3 Описание связей между составными частями программы	13
3.4 Связи программы с другими программами	14
4 Раздел 4. Используемые технические средства	15
4.1 Типы электронных вычислительных машин и устройств, которые используются при работе программы	15
5 Раздел 5. Входные данные	16
5.1 Формат входных данных	16
5.2 Описание входных данных	16
6 Раздел 6. Выходные данные	17
6.1 Формат выходных данных	17
6.2 Описание выходных данных	17

Заключение	18
Список использованной литературы	19

ВВЕДЕНИЕ

Написанная программа должна помочь сэкономить время и служить для автоматизации вывода и обработки данных с изображения. В качестве изображения используется фотография образца гербария с имеющейся на ней информационной этикеткой. Технология оптического распознавания переводит изображение в формат, доступный для редактирования. Программа находит буквы, объединяет их в слова и предложения, воссоздавая текст.

Остается только распределить полученную информацию в соответствии с ее значением. То есть распределить по категориям: 'Страна:', 'Семейство:', 'Род:', 'Название вида:', 'Внутренний номер:', 'Коллектор/-ы:', 'Место сбора:', 'Экология:', 'Высота н.у.м.:', 'Координаты сбора:', 'Дата сбора:', 'Автор определения:', 'Типовой статус:'.

Оптическое распознавание символов (англ. optical character recognition, OCR) — механический или электронный перевод изображений рукописного, машинописного или печатного текста в текстовые данные, использующиеся для представления символов в компьютере (например, в текстовом редакторе). Распознавание широко применяется для преобразования книг и документов в электронный вид.

Точное распознавание латинских символов в печатном тексте в настоящее время возможно, только если доступны чёткие изображения, такие, как сканированные печатные документы. Точность при такой постановке задачи превышает 99 %, абсолютная точность может быть достигнута только путём последующего редактирования человеком. Проблемы распознавания рукописного «печатного» и стандартного рукописного текста, а также печатных текстов других форматов (особенно с очень большим числом символов) в настоящее время являются предметом активных исследований. [2]

1. РАЗДЕЛ 1. ОБЩИЕ СВЕДЕНИЯ

1.1. Обозначение и наименование программы

Полное наименование - «Оптическое распознавание символов для «Virtual Herbarium ALTB» ».

Краткое наименование - « OCR Virtual Herbarium ALTB».

1.2. Программное обеспечение, необходимое для функционирования программы

Клиентская часть:

—операционная система Windows, MacOS, Linux, и т.д.

Программные средства внешних систем:

—язык программирования Python3.6 и выше;

—модули: OpenCV-python v. 4.5; EasyOCR v. 1.4.1; Pillow v. 8.2.0;

1.3. Язык программирования, на котором написана программа и используемые модули

Программа написана на языке Python3.6.9.

Для создания кроссплатформенной программы с графическим интерфейсом используется поставляемая с Python библиотека tkinter на основе Tcl/Tk.

Для возможности выполнять структурный анализ файлов CSV используется одноименный модуль csv. С его помощью происходит открытие/создание электронной таблицы для внесения в нее изменений.

Для извлечения текста из изображения и построение рамки вокруг него на фото используется EasyOCR. EasyOCR - это пакет Python, который позволяет преобразовывать изображение в текст.

Для загрузки изображений из файлов, создания новых изображений и отображения в интерфейсе программы используется библиотека Pillow. Библиотека изображений Python, или PIL (Python Imaging Library) нужна для обработки графики в Python.

2. РАЗДЕЛ 2. ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ

2.1. Классы решаемых задач

Классы решаемых задач:

- задачи извлечения информации;
- задачи визуализации информации;
- задачи обработки информации.

2.2. Назначение программы

Функциональное назначение:

- извлечение данных с изображений;
- наглядное графическое и табличное представление полученной информации;
- распределение, редактирование информации;
- сохранение информации записанной в формате .csv .

Эксплуатационное назначение:

- для использования в научных библиотеках.

3. РАЗДЕЛ 3. ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ

3.1. Используемые методы

3.1.1. Tkinter

class tkinter.ttk.Treeview

column(column, option=None, **kw): Изменение параметров для указанного column.

heading(column, option=None, **kw): Изменение параметров заголовка для указанного column.

delete(*items): Удалить все указанные items и все их потомки.

insert(parent, index, iid=None, **kw): Создает новый элемент и возвращает идентификатор вновь созданного элемента. parent - это идентификатор родительского элемента или пустая строка для создания нового элемента верхнего уровня. index - целое число или значение «END», указывающее место в списке для вставки нового элемента.

item(item, option=None, **kw): Изменение параметров для указанного item.

selection(): Возвращает кортеж выбранных элементов.

class tkinter.ttk.Style Класс используемый для управления базой данных стилей.

configure(style, query_opt=None, **kw): Задаёт значение по умолчанию для указанных параметров в style.

map(style, query_opt=None, **kw): Задание динамических значения указанных параметров в style.

xscrollcommand - Используется для связи с горизонтальными полосами прокрутки. **yscrollcommand** - Используется для связи с вертикальными полосами прокрутки.

Модули, обеспечивающие поддержку Tk, включают:

tkinter.filedialog Общие диалоговые окна, позволяющие пользователю указать файл для открытия или сохранения.

asksaveasfile

askopenfilename

tkinter.messagebox

showinfo Доступ к стандартным диалоговым окнам Tk. [5]

3.1.2. OpenCV

`cv.imread(filename[, flags])`: Функция `imread` загружает изображение из указанного файла и возвращает его. Если изображение не может быть прочитано (из-за отсутствия файла, неправильных разрешений, неподдерживаемого или недопустимого формата), функция возвращает пустую матрицу.

`cv.cvtColor(src, code[, dst[, dstCn]])`: Функция преобразует входное изображение из одного цветового пространства в другое. В случае преобразования цветового пространства в цветовое пространство RGB, порядок каналов должен быть указан явно (RGB или BGR).

`cv.rectangle(img, pt1, pt2, color[, thickness[, lineType[, shift]])`: Функция `cv.rectangle` рисует контур прямоугольника или прямоугольник с заливкой, два противоположных угла которого - `pt1` и `pt2`. Возвращает измененное изображение.

`cv.putText(img, text, org, fontFace, fontScale, color[, thickness[, lineType[, bottomLeftOrigin]])`: Функция `cv.putText` отображает указанную текстовую строку в изображении. [3]

3.1.3. Pillow

`PIL.Image.open(fp, mode='r', formats=None)`: Открывает и идентифицирует указанный файл изображения.

`PIL.ImageTk.PhotoImage(image=None, size=None, **kw)`: Преобразует изображение в фотоизображение, совместимое с Tkinter. [4]

3.1.4. Csv

`csv.writer(csvfile, dialect='excel', **fmtparams)` Метод `csv.writer` возвращает объект записи, преобразует данные в строки с разделителями для данного CSV-файла.

`csvwriter.writerow(row)` Метод `writerow` записывает строку данных в указанный CSV-файл.

`csv.reader(csvfile, dialect='excel', **fmtparams)` Метод `csv.reader` считывает значения по строкам в данном CSV-файле. [1]

3.1.5. EasyOCR

`reader.readtext` Функция чтения текста. При запуске данного метода в виде списка возвращаются проанализированные строки, координаты текстовой области, а также уровень уверенности. [EasyOCR]

3.2. Описание функций составных частей программы

3.2.1. Функции главной программы

1. `def func_fill_tree(value='-')`:
2. `def select_all()`:
3. `def func_add_record()`:
4. `def func_remove_all()`:
5. `def func_select(event)`:
6. `def func_update()`:
7. `def func_save()`:
8. `def func_insert()`:
9. `def select_file(choice)`:
10. `def func_view_photo(img)`:
11. `def func_view_csv(filename)`:
12. `def func_scan_img(filename_img)`:

3.2.2. Функции подмодуля программы

1. `def main(filename)`:
2. `class InfoString`:
 - 2.1. `def __init__(self, bbox, text)`:
 - 2.2. `def truncate(self, img)`:
 - 2.3. `@classmethod def text(cls)`:

3.3. Описание связей между составными частями программы

В главную программу импортирован модуль подпрограммы. Название подпрограммы является её идентификатором, через который получается доступ к атрибутам, определенным внутри неё.

Доступ к атрибутам модуля осуществляется с помощью точечной нотации. Так для вызова подпрограммы из главной используется конструкция:

```
info_list, img = module.main(filename_img);
```

На вход функции `main()` подпрограммы `module` поступает `filename_img`, содержащий в себе полный путь для доступа к файлу с изображением.

На выход из функции поступает `info_list` и `img`, которые записываются в одноименные переменные в главной программе. Теперь `info_list` содержит в себе список из строк, полученных с изображения, а `img` обработанное изображение.

3.4. Связи программы с другими программами

С программой используется встроенное средство выбора файлов.

В средстве выбора файлов отображаются сведения, позволяющие сориентировать пользователя и обеспечить ему привычное взаимодействие с системой при открытии и сохранении файлов.

4. РАЗДЕЛ 4. ИСПОЛЬЗУЕМЫЕ ТЕХНИЧЕСКИЕ СРЕДСТВА

4.1. Типы электронных вычислительных машин и устройств, которые используются при работе программы

Программа совместима с операционной системой Windows и Linux, при условии установленных модулей используемых в ней.

5. РАЗДЕЛ 5. ВХОДНЫЕ ДАННЫЕ

5.1. Формат входных данных

На вход могут поступать файлы в формате .jpg/ .png изображений и .csv табличных данных.

5.2. Описание входных данных

Изображение поступающее на вход программы должно быть хорошего разрешения для наиболее точного получения реального текста.

В файле с табличными данными .csv должно быть использовано разделение столбцов данных по ”,”.

6. РАЗДЕЛ 6. ВЫХОДНЫЕ ДАННЫЕ

6.1. Формат выходных данных

Выходные данные, представляют собой текстовые файлы в формате .csv.

6.2. Описание выходных данных

Тип файла, имеющий расширение CSV, содержит информацию, которую могут импортировать в базу данных, так же подобный текстовый файл может носить данные, необходимые для организации таблиц.

ЗАКЛЮЧЕНИЕ

Написанная программа сокращает время затрачиваемое на занесение данных о гербарных образцах в электронную базу данных. Упрощает процесс занесения информации в таблицу, делая его интерактивным и быстрым.

Данную программу можно использовать в научных библиотеках, в учебных заведениях или в собственных целях.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. [Электронный ресурс] CSV File Reading and Writing. — URL: <https://docs.python.org/3/library/csv.html>.
2. [Электронный ресурс] OCR - Википедия. — URL: <https://ru.wikipedia.org/wiki/>.
3. [Электронный ресурс] OpenCV - Image file reading and writing. — URL: https://docs.opencv.org/3.4/d4/da8/group__imgcodecs.html#ga288b8b3da0892bd651fce07b3bbd3a56.
4. [Электронный ресурс] Reference — Pillow. — URL: <https://pillow.readthedocs.io/en/stable/reference/index.html>.
5. [Электронный ресурс] Tk — Commands. — URL: <https://www.tcl.tk/man/tcl8.6/TkCmd/contents.html>.