Mini Project(2020-2021)

AUTOMATIC CERTIFICATE GENERATOR

Using

(OPENCV AND PYTHON)

Department Of Computer Engineering&Applications

Submitted By:

NITIN KUMAR SINGH(181500434)

NIKHIL SINGHAL(181500429)

AMIT VARSHNEY(181500087)

Submitted To:

PRIYA AGARWAL

(TECHNICAL TRAINER)

TABLE OF CONTENTS

Certificates Of Course Completion For This Projects

Nitin Kumar Singh's Certificate

Nikhil Singhal's Certificate

Amit Varshney's Certificate

Department of computer Engineering and Applications
GLA University, Mathura
17 km. Stone NH#2, Mathura-Delhi Road, P.O. – Chaumuha,
Mathura – 281406

Project Information:

Title Of Project/Training/Task	PYTHON AUTOMATIC GENRATOR
Role & Responsibility	Nitin KumarSingh NumPy library in python Nikhil singhal-os,csv in python Amit Varshney-PIL,OpenCV
Technical Details	**Hardware Requirements:** Processor Used: Intel i3 or above Operating System: Windows 7 or above RAM :4GB or above Hardware Devices: Computer System Hard Disk:256GB or above Software Requirements: Technology used: Python idle Language used:Python3

Training Implementation Details	Fully Implemented
Training Period	FROM 18-AUG-2020 TO 25-NOV-2020

Department of computer Engineering and Applications

GLA University, Mathura

17 km. Stone NH#2, Mathura-Delhi Road, P.O.– Chaumuha, Mathura – 281406

Declaration

This is certify that Nitin Kumar Singh,Nikhil Singhal and Amit Varshney student of B.tech(CSE) 3rd year has successfully Completed the MINI PROJECT named AUTOMATIC CERTIFICATE GENERATOR under the Guidance of Priya Agarwal During 2020-21.

Signature:

Priya Agarwal(Mentor)

ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the synopsis of the B.Tech Mini Project

(AUTOMATIC CERTIFICATE GENERATOR)

Undertaken during B.Tech 3rd year. This project in itself is going to acknowledgement to the inspiration, drive and technical assistance will be contributed to it by many individuals.

We owe special debt of gratitude to Priya Agrawal,Training And Development for providing with an encouraging platform to develop this Project, which thus helped us in shaping our abilities towards a constructive goal and for his constant support and guidelines to our work. His sincerity, thoroughness and perseverance is been a constant source of motivation for us. We believe that he will shower us with all his extensively experienced ideas and insightful comments at different stages of project and taught us about the latest industry-oriented technologies.

We also do not like to miss the opportunity to acknowledge the contribution of all faculty members of department for their guidance and contribution.

1. Nitin Kumar Singh(181500434)
2. Nikhil Singhal(181500429)
3. Amit Varshney(181500087)

ABSTRACT

In this using python script automates the process of generating multiple certificates for a given list of people from a template image file. this script will take care of the generation of certificate for different people using txt file as an input of names and coordinates of template.To explain it in a better way, say you need to provide digital certificates for any purpose, so create a template for a certificate. It autogenerates the certificates of any event of the individual candidates. Generating certificate of participation of each attende can be cumbersome and hard work. Automating this job can easily save tons of time and manual work. We have made this automation possible and quicker than ever. This script can makes thousand of certificates with as simple input as just a name. Our algorithm's quickest runtime is just seconds to generate a single certificate and with minimum possible memory consumption.

INTRODUCTION

Overview

By applying the python module and libraries we produce the template of certificate consists of name of participant and date of event.

We apply python technology we produce certificate by just working on saved .txt files in system consists of name.

This project automatic certificate generator using python and OpenCV implements a classic version of

Certificate generator. In this the python script automates the process of generating multiple certificates for a given list of people from a template image file. To explain it in a better way, say you need to provide digital certificates for any purpose, so you create a template for a certificate, this script will take care of the generation of certificate for different people using txt file as an input of names and we also modify coordinate of another txt file.This certificate generator includes two.txt files one for name of the candidate with date and another for co-ordinates files. after all this we run the

python code and it will fetch all the details from these .text files and will generate our certificate.

AREAS OF COMPUTER SCIENCE:

1- COMPUTER VISION

2-PROGRAMMING LANGUAGES

3-COMPUTER GRAPHICS

4-NUMERIC ANALYSIS

HARDWARE REQUIREMENTS:

- PROCESSOR USED:-Intel Pentium or above
- OPERATING SYSTEM:-Win 7 or above
- RAM:-4GB or above
- HARDWARE DEVICES:-Computer or Laptop System
- HARD DISK:-256GB or above

SOFTWARE REQUIREMENTS:

- TECHNOLOGY USED : OpenCV
- LANGUAGE USED:PYTHON3

PROBLEM DEFINATION:

I am a core-committee member of the University Coding Club, and hence we organize many competitions, workshops or events where we need to provide a certificate of participation to the users. So we require to generate large amount of certificates for this competitions .

OBJECTIVE:

Automatic Certificate Generator aims to generate the large amount of digital certificates for many competitions, workshops and events.

It also aims to ease the manual work as a university or other institute members to distribute certificates to the participants using python script(Open CV) that automates the process of generating certificates from given

template.Simply edit the input and output file for the CSV input and template image and the generated certificates files and run it.

SOFTWARE DESIGN

A software design document (SDD) is a written description of a software product, that a software designer writes in order to give a software development team overall guidance to the architecture of the software project. An SDD usually accompanies an architecture diagram with pointers to detailed feature specifications of smaller pieces of the design. Practically, a design document is required to coordinate a large team under a single vision. A design document needs to be a stable reference, outlining all parts of the software and how they will work. The document is commanded to give a fairly complete description, while maintaining a high-level view of the software.

Testing

Introduction:

The implementation phase of software development is concerned with translating design specification into source code. The preliminary goal of implementation is to write source code and internal documentation so that conformance of the code to its specifications can be easily verified, and so that debugging, testing and modifications are eased. This goal can be achieved by making the source code as clear and straightforward as possible. Simplicity, clarity and elegance are the hallmark of good programs, obscurity, cleverness, and complexity are indications of inadequate design and misdirected thinking.

Source code clarity is enhanced by structured coding techniques, by good coding style, by, appropriate supporting documents, by good internal comments, and by feature provided in modern programming languages.

Terms in Testing Fundamental:

1. Error

2. Fault

3. Failure

4. Functional Test

Functional test cases involve exercising the code with Nominal input values for which expected results are known; as well as boundary values (minimum values, maximum values and values on and just outside the functional boundaries) and special values.

5. Performance Test
Performance testing determines the amount of execution time spent in various parts of the unit, program throughput, response time, and device utilization by the program unit.

6. Stress Test
Stress test are those designed to intentionally break the unit. A great deal can be learned about the strengths and limitations of a program by examining the manner in which a program unit breaks.

7. Structure Test

Structure tests are concerned with exercising the internal logic of a program and traversing particular execution paths. Some authors refer collectively to functional performance and stress testing as “black box” testing

Test Cases	Description	Prediction	Result
1	Import required modules and Packages	Installed	OK
2	Save two .txt file	Saved Successfully	OK
3	Download template For certificate	download successfully	OK
4	Certificate appear With date and coordinate	Run successfully	OK

METHODOLOGY:

Any event usually involves a lot of participants and generating handwritten certificates for each one of them and sending them digitally is a really tedious task. Automating this job can easily save tons of time and manual work and thus also reducing the error rate. So we use the concept of python with OpenCV which will generates the certificates in a very less time and also Python script generates certificates with the person's name, reading from an .txt file after loading a template certificate in the script.

IMPLEMENTATION DETAILS:-

We make this project by making two sets which is set coordinates and generate certificate so in set coordinate we set the coordinates of name and date and in case pf generate certificate we select text file of names and also change default date and generate multiple certificates.

And also we required to implement various libraries such as PIL,OpenCV,csv,os,NumPy. This libraries played an important role for automation of certificates.

PIL(Python Image Library) is used Image font,Image draw and Image.it is used for opening, manipulating and saving many different image file formats.

OpenCV is the huge open-source library for computer vision, machine learning, and image processing and now it plays a

major role in real-time operation which is very important in today's systems. By using it, one can process images and videos to identify objects, faces, or even the handwriting of a human.

The CSV module implements classes to read data in CSV format. It allows programmers to say, "write this data in the format.. the CSV formats understood by other applications or define their own special-purpose CSV formats.

The OS module allows interaction with the Operating System, either generically or specific to a particular OS.

NUMPY is a Python package. It stands for 'Numerical Python'. It is a library consisting of multidimensional array objects and a collection of routines for processing of array.

Code Of Project

```
from PIL import ImageFont, ImageDraw, Image
import cv2
import numpy as np
import os
import csv

f = open("names.txt","r")
names_list = f.read().split("\n")
#print(names_list)

f1 = open("coords.txt","r")
coordinates = f1.read().split("\n")
```

```
flag=True

for i in range(len(names_list)):
name_to_print = names_list[i]
date_to_print = "20/11/2020"#Change this date as per requirement

# Load image in OpenCV
image = cv2.imread("ce3.jpg")

# Convert the image to RGB (OpenCV uses BGR)
cv2_im_rgb = cv2.cvtColor(image,cv2.COLOR_BGR2RGB)

# Pass the image to PIL
pil_im = Image.fromarray(cv2_im_rgb)

draw = ImageDraw.Draw(pil_im)
# use a truetype font
```

```
font = ImageFont.truetype("./fonts/MATURASC.TTF", 29)
#You can change fonts from list given bottom
font1 = ImageFont.truetype("./fonts/OLDENGL.TTF", 22)



# Draw the text
draw.text((int(coordinates[0]), int(coordinates[1])),
name_to_print, font=font , fill='black')
draw.text((int(coordinates[2]), int(coordinates[3])),
date_to_print , font=font1, fill='blue')


# Get back the image to OpenCV
cv2_im_processed = cv2.cvtColor(np.array(pil_im),
cv2.COLOR_RGB2BGR)


if flag:
cv2.imshow('Certificate', cv2_im_processed) #Shows sample
image
flag=False
path = ''
```

```
cv2.imwrite('./output/'+name_to_print+'.png',cv2_im_processed)

#os.startfile('output.png')

cv2.waitKey(0)

cv2.destroyAllWindows()
```

CONTRIBUTION SUMMARY:

This mini project " AUTOMATIC CERTIFICATE GENERATOR" with the help of Python and OpenCV

It eas not made by an individual person.it will make by equal contribution of our group members we divided the task into groups so we will make project effectively and some new ideas come and implements on this project.

NITIN KUMAR SINGH-NUMPY library and their working

NIKHIL SINGHAL-OS And CSV MODULE and their working

AMIT VARSHNEY-OPENCV And PIL LIBRARY and their working.

PROJECT WORK

SOURCE CODE WITH OUTPUT SCREENSHOTS

certificate.py - D:\miniprojectimplementation\Certificate-Automation\certificate.py (3.8.0)

File Edit Format Run Options Window Help

```
from PIL import ImageFont, ImageDraw, Image
import cv2
import numpy as np
import os
import csv

f = open("names.txt","r")
names_list = f.read().split("\n")
#print(names_list)

f1 = open("coords.txt","r")
coordinates = f1.read().split("\n")

flag=True

for i in range(len(names_list)):
    name_to_print = names_list[i]
    date_to_print = "21/11/2020"#Change this date as per requirement

    # Load image in OpenCV
    image = cv2.imread("ce3.jpg")

    # Convert the image to RGB (OpenCV uses BGR)
    cv2_im_rgb = cv2.cvtColor(image,cv2.COLOR_BGR2RGB)

    # Pass the image to PIL
    pil_im = Image.fromarray(cv2_im_rgb)

    draw = ImageDraw.Draw(pil_im)
    # use a truetype font
    font = ImageFont.truetype("./fonts/copperplate gothic font.TTF", 29)      #You can change fonts from list given bottom
    font1 = ImageFont.truetype("./fonts/OLDENGL.TTF", 22)

    # Draw the text
    draw.text((int(coordinates[0]), int(coordinates[1])), name_to_print, font=font , fill='black')
    draw.text((int(coordinates[2]), int(coordinates[3])), date_to_print , font=font1, fill='blue')

    # Get back the image to OpenCV
    cv2_im_processed = cv2.cvtColor(np.array(pil_im), cv2.COLOR_RGB2BGR)

    if flag:
        cv2.imshow('Certificate', cv2_im_processed) #Shows sample image
        flag=False
    path = ''
    cv2.imwrite('./output/'+name_to_print+'.png',cv2_im_processed)
    #os.startfile('output.png')
```

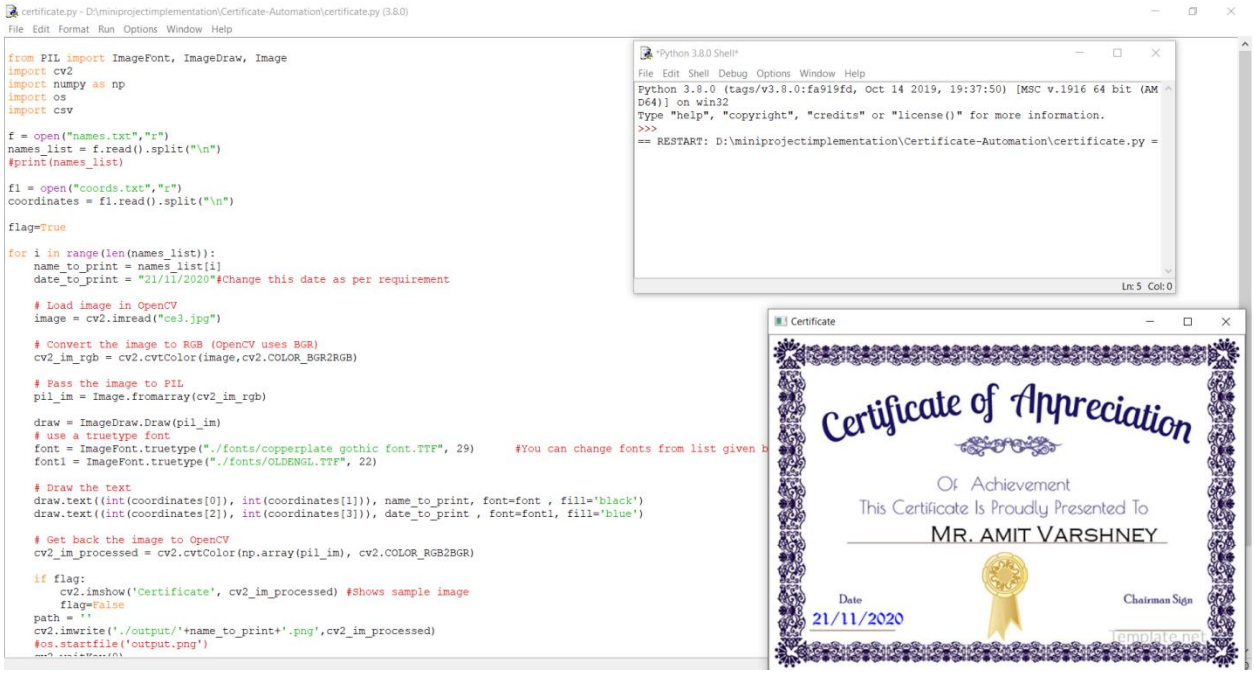
Ln: 1 Col: 0

certificate.py - D:\miniprojectimplementation\Certificate-Automation\certificate.py (3.8.0)

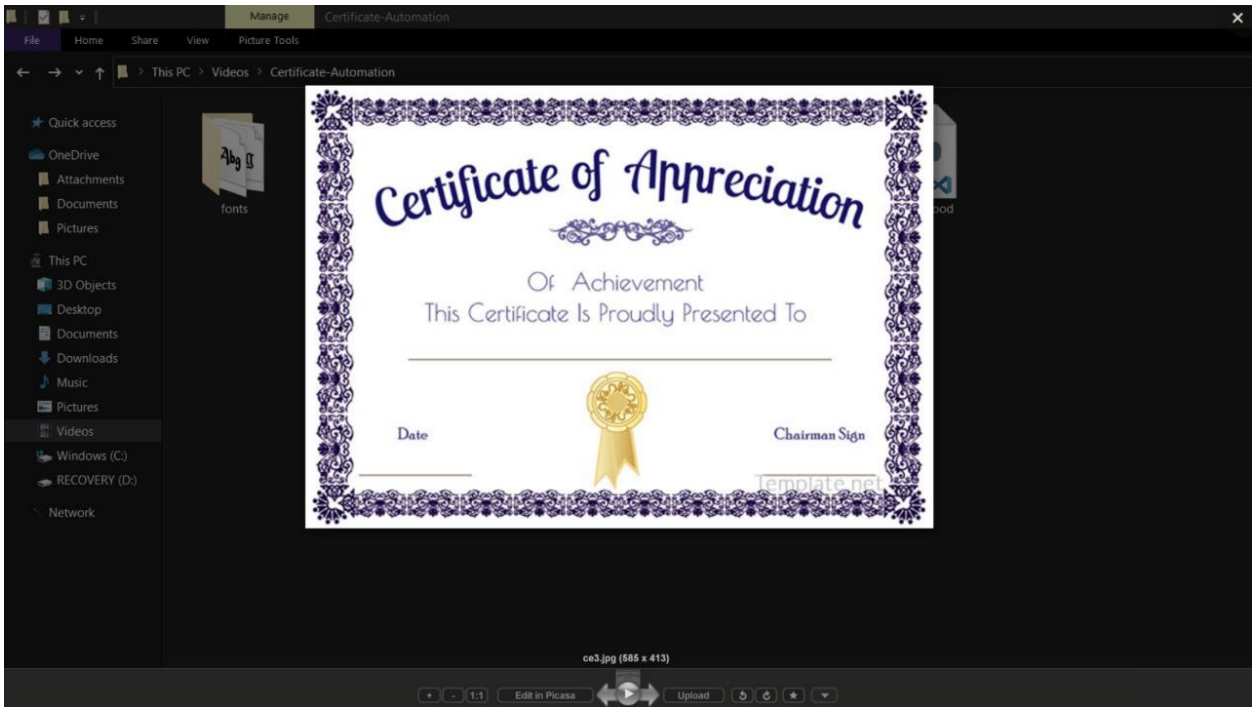
File Edit Format Run Options Window Help

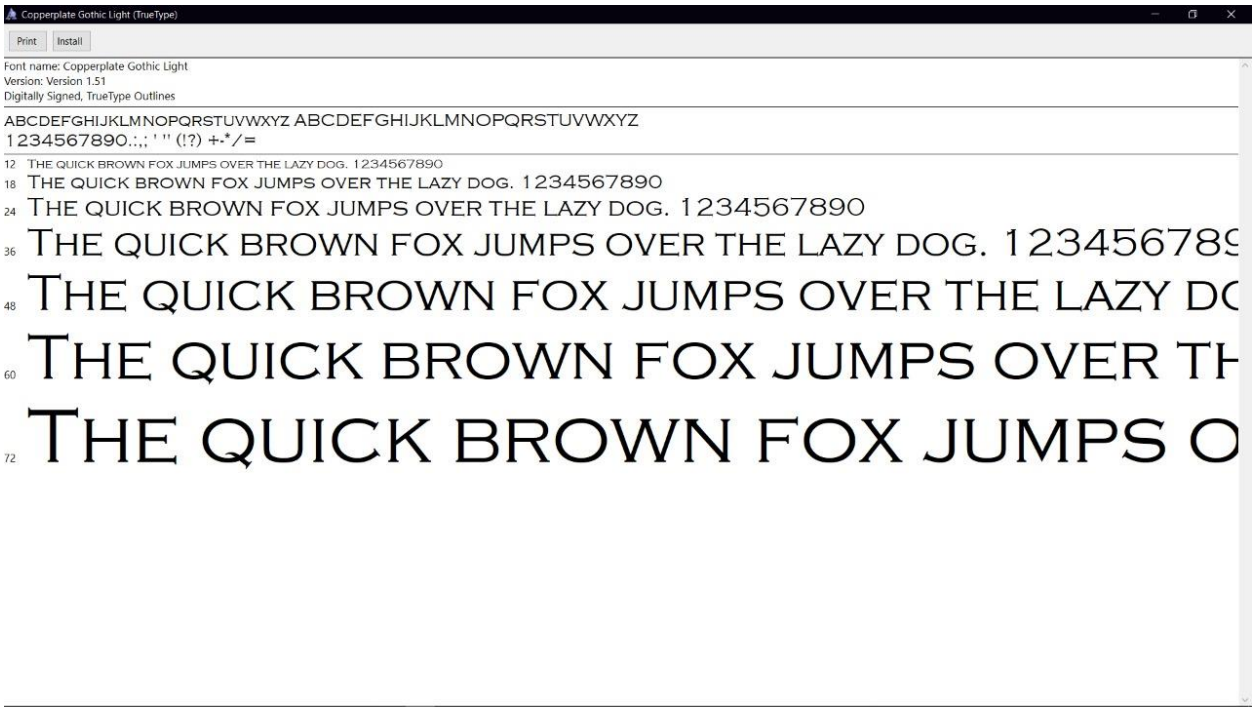
```
import os
import csv

f = open("names.txt","r")
names_list = f.read().split("\n")
#print(names_list)

f1 = open("coords.txt","r")
coordinates = f1.read().split("\n")

flag=True

for i in range(len(names_list)):
    name_to_print = names_list[i]
    date_to_print = "21/11/2020"#Change this date as per requirement

    # Load image in OpenCV
    image = cv2.imread("ce3.jpg")

    # Convert the image to RGB (OpenCV uses BGR)
    cv2_im_rgb = cv2.cvtColor(image,cv2.COLOR_BGR2RGB)

    # Pass the image to PIL
    pil_im = Image.fromarray(cv2_im_rgb)

    draw = ImageDraw.Draw(pil_im)
    # use a truetype font
    font = ImageFont.truetype("./fonts/copperplate gothic font.TTF", 29)      #You can change fonts from list given bottom
    font1 = ImageFont.truetype("./fonts/OLDENGL.TTF", 22)

    # Draw the text
    draw.text((int(coordinates[0]), int(coordinates[1])), name_to_print, font=font , fill='black')
    draw.text((int(coordinates[2]), int(coordinates[3])), date_to_print , font=font1, fill='blue')

    # Get back the image to OpenCV
    cv2_im_processed = cv2.cvtColor(np.array(pil_im), cv2.COLOR_RGB2BGR)

    if flag:
        cv2.imshow('Certificate', cv2_im_processed) #Shows sample image
        flag=False
    path = ''
    cv2.imwrite('./output/'+name_to_print+'.png',cv2_im_processed)
    #os.startfile('output.png')
    cv2.waitKey(0)

    cv2.destroyAllWindows()
```

FINAL OUTPUT

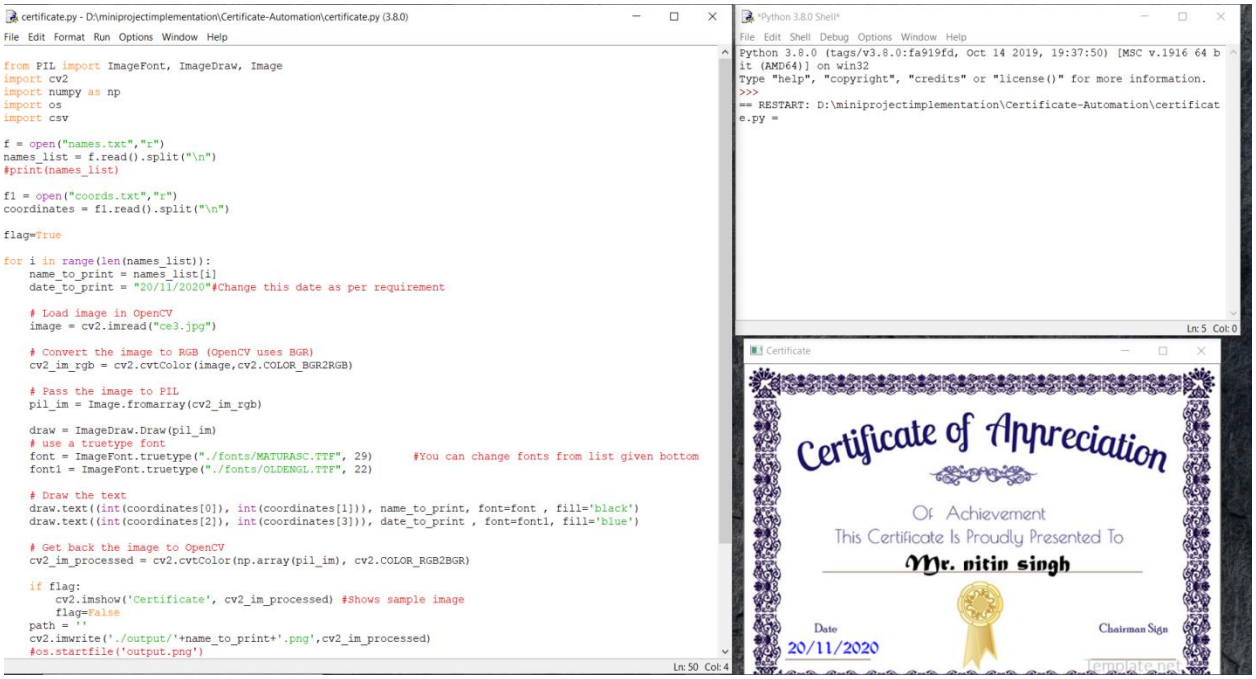


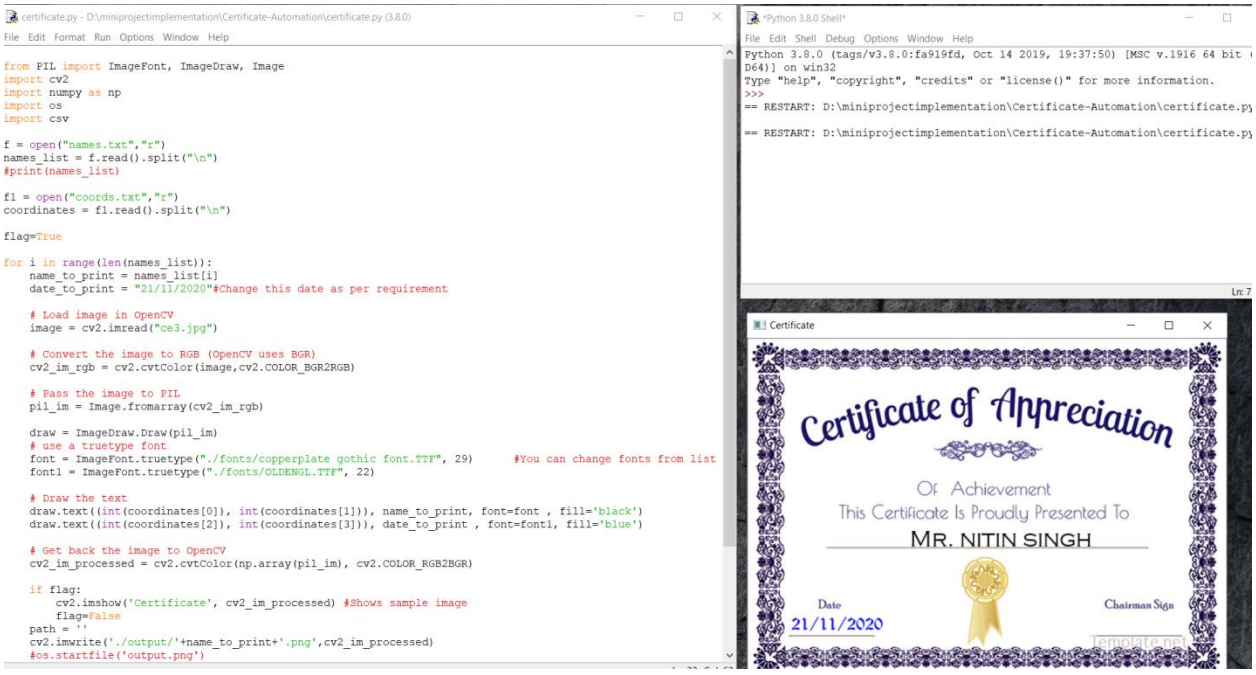
SCREENSHOTS

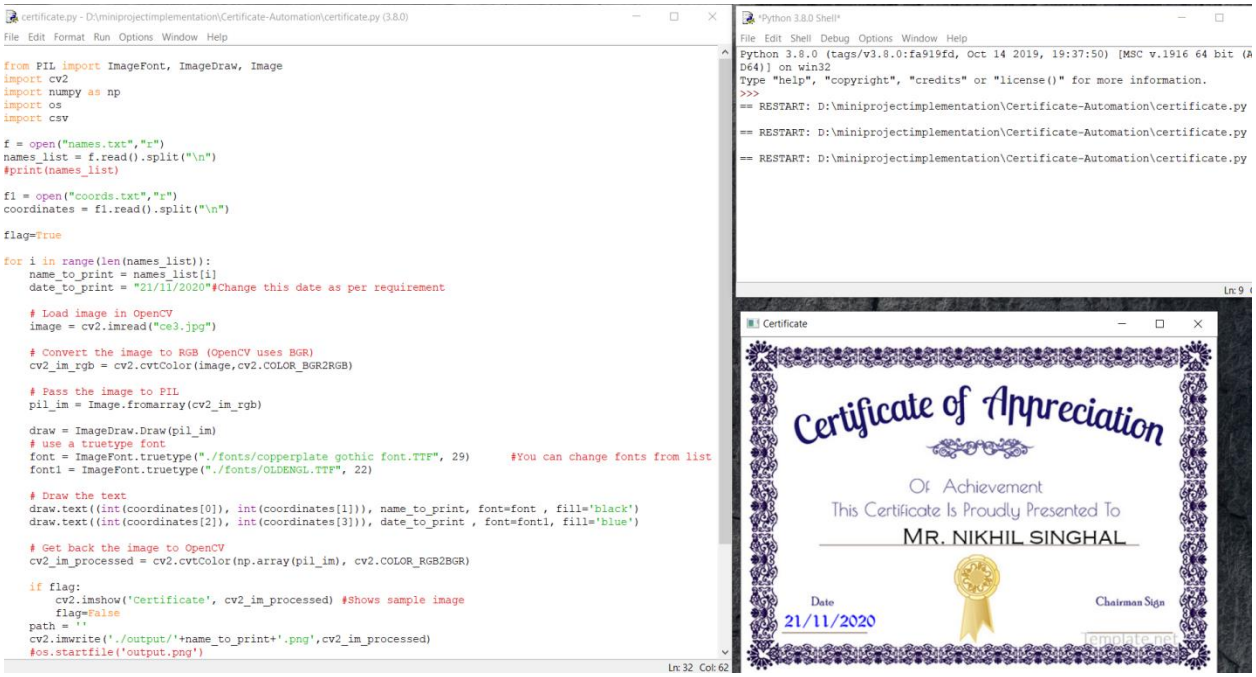

Manage
Certificate-Automation
File
Home
Share
View
Picture Tools
This PC
Videos
Certificate-Automation
Quick access
OneDrive
Attachments
Documents
Pictures
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Windows (C:)
RECOVERY (D:)
Network
fonts
Certificate of Appreciation
Of Achievement
This Certificate Is Proudly Presented To
Date
Chairman Sign
Template.net
ce3.jpg (585 x 413)
1:1
Edit in Picasa
Upload


Copperplate Gothic Light (TrueType)
Print
Install
Font name: Copperplate Gothic Light
Version: Version 1.51
Digitally Signed, TrueType Outlines
ABCDEFGHIJKLMNOPQRSTUVWXYZ ABCDEFGHIJKLMNOPQRSTUVWXYZ
1234567890.:,; ' " (!?) +-*/=
12 THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG. 1234567890
18 THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG. 1234567890
24 THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG. 1234567890
36 THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG. 123456789
48 THE QUICK BROWN FOX JUMPS OVER THE LAZY DO
60 THE QUICK BROWN FOX JUMPS OVER TH
72 THE QUICK BROWN FOX JUMPS O

Screenshots With Different Outputs




certificate.py - D:\miniprojectimplementation\Certificate-Automation\certificate.py (3.8.0)
File Edit Format Run Options Window Help
from PIL import ImageFont, ImageDraw, Image
import cv2
import numpy as np
import os
import csv
f = open("names.txt","r")
names_list = f.read().split("\n")
#print(names_list)
f1 = open("coords.txt","r")
coordinates = f1.read().split("\n")
flag=True
for i in range(len(names_list)):
name_to_print = names_list[i]
date_to_print = "21/11/2020"#Change this date as per requirement
# Load image in OpenCV
image = cv2.imread("ce3.jpg")
# Convert the image to RGB (OpenCV uses BGR)
cv2_im_rgb = cv2.cvtColor(image,cv2.COLOR_BGR2RGB)
# Pass the image to PIL
pil_im = Image.fromarray(cv2_im_rgb)
draw = ImageDraw.Draw(pil_im)
# use a truetype font
font = ImageFont.truetype("./fonts/copperplate gothic font.TTF", 29) #You can change fonts from list
font1 = ImageFont.truetype("./fonts/OLDENGL.TTF", 22)
# Draw the text
draw.text((int(coordinates[0]), int(coordinates[1])), name_to_print, font=font , fill='black')
draw.text((int(coordinates[2]), int(coordinates[3])), date_to_print , font=font1, fill='blue')
# Get back the image to OpenCV
cv2_im_processed = cv2.cvtColor(np.array(pil_im), cv2.COLOR_RGB2BGR)
if flag:
cv2.imshow('Certificate', cv2_im_processed) #Shows sample image
flag=False
path = ''
cv2.imwrite('./output/'+name_to_print+'.png',cv2_im_processed)
#os.startfile('output.png')
Python 3.8.0 Shell
File Edit Shell Debug Options Window Help
Python 3.8.0 (tags/v3.8.0:fa919fd, Oct 14 2019, 19:37:50) [MSC v.1916 64 bit (
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
== RESTART: D:\miniprojectimplementation\Certificate-Automation\certificate.py
== RESTART: D:\miniprojectimplementation\Certificate-Automation\certificate.py
Ln: 7
Certificate
Certificate of Appreciation
Of Achievement
This Certificate Is Proudly Presented To
MR. NITIN SINGH
Date
21/11/2020
Chairman Sign


certificate.py - D:\miniprojectimplementation\Certificate-Automation\certificate.py (3.8.0)
File Edit Format Run Options Window Help
from PIL import ImageFont, ImageDraw, Image
import cv2
import numpy as np
import os
import csv
f = open("names.txt","r")
names_list = f.read().split("\n")
#print(names_list)
f1 = open("coords.txt","r")
coordinates = f1.read().split("\n")
flag=true
for i in range(len(names_list)):
name_to_print = names_list[i]
date_to_print = "21/11/2020"#Change this date as per requirement
# Load image in OpenCV
image = cv2.imread("ce3.jpg")
# Convert the image to RGB (OpenCV uses BGR)
cv2_im_rgb = cv2.cvtColor(image,cv2.COLOR_BGR2RGB)
# Pass the image to PIL
pil_im = Image.fromarray(cv2_im_rgb)
draw = ImageDraw.Draw(pil_im)
# use a truetype font
font = ImageFont.truetype("./fonts/copperplate gothic font.TTF", 29) #You can change fonts from list
font1 = ImageFont.truetype("./fonts/OLDENGL.TTF", 22)
# Draw the text
draw.text((int(coordinates[0]), int(coordinates[1])), name_to_print, font=font , fill='black')
draw.text((int(coordinates[2]), int(coordinates[3])), date_to_print , font=font1, fill='blue')
# Get back the image to OpenCV
cv2_im_processed = cv2.cvtColor(np.array(pil_im), cv2.COLOR_RGB2BGR)
if flag:
cv2.imshow('Certificate', cv2_im_processed) #Shows sample image
flag=False
path = ''
cv2.imwrite('./output/'+name_to_print+'.png',cv2_im_processed)
#os.startfile('output.png')
Ln: 32 Col: 62
Python 3.8.0 Shell
File Edit Shell Debug Options Window Help
Python 3.8.0 (tags/v3.8.0:fa919fd, Oct 14 2019, 19:37:50) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
== RESTART: D:\miniprojectimplementation\Certificate-Automation\certificate.py
== RESTART: D:\miniprojectimplementation\Certificate-Automation\certificate.py
== RESTART: D:\miniprojectimplementation\Certificate-Automation\certificate.py
Ln: 9
Certificate
Certificate of Appreciation
Of Achievement
This Certificate Is Proudly Presented To
MR. NIKHIL SINGHAL
Date
21/11/2020
Chairman Sign

certificate.py - D:\miniprojectimplementation\Certificate-Automation\certificate.py (3.8.0)

File Edit Format Run Options Window Help

```
from PIL import ImageFont, ImageDraw, Image
import cv2
import numpy as np
import os
import csv

f = open("names.txt","r")
names_list = f.read().split("\n")
#print(names_list)

f1 = open("coords.txt","r")
coordinates = f1.read().split("\n")

flag=True

for i in range(len(names_list)):
    name_to_print = names_list[i]
    date_to_print = "21/11/2020"#Change this date as per requirement

    # Load image in OpenCV
    image = cv2.imread("ce3.jpg")

    # Convert the image to RGB (OpenCV uses BGR)
    cv2_im_rgb = cv2.cvtColor(image,cv2.COLOR_BGR2RGB)

    # Pass the image to PIL
    pil_im = Image.fromarray(cv2_im_rgb)

    draw = ImageDraw.Draw(pil_im)
    # use a truetype font
    font = ImageFont.truetype("./fonts/copperplate gothic font.TTF", 29)      #You can change fonts from list
    font1 = ImageFont.truetype("./fonts/OLDENGL.TTF", 22)

    # Draw the text
    draw.text((int(coordinates[0]), int(coordinates[1])), name_to_print, font=font , fill='black')
    draw.text((int(coordinates[2]), int(coordinates[3])), date_to_print , font=font1, fill='blue')

    # Get back the image to OpenCV
    cv2_im_processed = cv2.cvtColor(np.array(pil_im), cv2.COLOR_RGB2BGR)

    if flag:
        cv2.imshow('Certificate', cv2_im_processed) #Shows sample image
        flag=False
    path = ''
    cv2.imwrite('./output/'+name_to_print+'.png',cv2_im_processed)
    #os.startfile('output.png')
```

Python 3.8.0 Shell

File Edit Shell Debug Options Window Help

```
Python 3.8.0 (tags/v3.8.0:fa919fd, Oct 14 2019, 19:37:50) [MSC v.1916 64 bit (A
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
== RESTART: D:\miniprojectimplementation\Certificate-Automation\certificate.py

== RESTART: D:\miniprojectimplementation\Certificate-Automation\certificate.py

== RESTART: D:\miniprojectimplementation\Certificate-Automation\certificate.py

== RESTART: D:\miniprojectimplementation\Certificate-Automation\certificate.py
```

Ln: 11

Certificate

FUTURE OF PROJECT:

In present scenario there are various competitions and events is held so it is very difficult task . Any event usually involves a lot of participants and generating handwritten certificates for each one of them and sending them digitally is a really tedious task. Automating this job can easily save tons of time and manual work and thus also reducing the error rate. This Python script generates certificates with the persons name, reading from an .txt file after loading a template certificate in the script.

Developed this project as mini project to ease the manual work as a university or other institute members to distribute certificates to the participants using python script(OpenCV) that automates the process of generating digital certificates from given template.it will reduce time and effective method for producing certificates.

->REFERENCES:

- www.youtube.com
- www.geeeksforgeeks.com
- www.w3school.com
- www.tutorialspoint.com
- www.learnpython.org

->BOOK REFERENCES

- Head-First Python(Paul Barry)
- Learn Python the Hard Way(Zed A.Shaw)
- Learning Python (David Ascher and Mark Lutz)

->FACULTY GUIDELINE:

Miss. PRIYA AGRAWAL

(TECHNICAL TRAINER)

Thank You