Final project image processing

Student: Diya Bakri ID:

Student: Mohammad Khader Zatari ID:

The following code extracts fragments of a torn paper in given in jpg pic:

**Files:**

FileReader.py:

The class file for reading images from a given file path with an image name or a wild card for reading multiple images returns a NumPy Array of images

SegPrep.py:

Contains all the used functions for extraction the constructor takes in as an input an image array and the object perform operations on the images in the list and saves the results in an attribute called prevResults

projectM2022\_q1.py:

the main file for extracting the bounding boxes of each element found in the image list in the form of an excel sheet for each image

projectM2022\_q2.py:

the main file for extracting the contours boxes of each element found in the image list in the form of an excel sheet for each image

**instruction to run the code:**

unzip the main folder, then create a results, excel folder or unzip the other folder.

To run projectM2022\_q1.py:

Open terminal and run python projectM2022\_q1.py and wait until get the photos in the results folder and it will pop up on the screen, when you close the first photo, the second photo will appear and so on.

To run projectM2022\_q2.py:

Open terminal and run python projectM2022\_q2.py and wait until get the photos in the results folder and it will pop up on the screen, when you close the first photo, the second photo will appear and so on.

To change path or the photos that we want to run the code on it, we just modify the

FileReader("./images", "\*.jpg"), the first parameter is the path to the folder, the second parameter is the requested photos

**Intro:**

In this project we are trying to contour the small pieces of the papers. we have a couple obstacles, one of the obstacles that we have a high diversity of photos, different contrasts, different pieces sizes.

**algorithm Flow:**

1. Gaussian smoothing.
2. Color quantization.
3. Morphological edge detection.
4. Finding borders.
5. Clearing outer parameter.
6. Reginal filling
7. clearing unwanted regions.
8. Labeling regions.
9. Extract each region bounding box.
10. Extract contours

**Color quantization:**

We created a method for segmenting the colors of an image by “n” equal segments such that 1 <= n <= 255 and each color that falls in a region of the segments are all given the same value,

This method has the most effect on our results we apply the quantization in a reverse pyramid shape which gives a result similar to that of K-mean the pyramid we used contained the values [100,50,25,10,8,5,4]

Example of running this function:

Original image:

Diagram

Description automatically generated with medium confidence

N = 100:

Diagram

Description automatically generated with low confidence

N = 50:

Diagram

Description automatically generated with medium confidence

N=25:

A picture containing diagram

Description automatically generated

N = 10:

A picture containing text

Description automatically generated

N = 8:

A picture containing text

Description automatically generated

N = 5:

Diagram

Description automatically generated with medium confidence

N = 4:

A picture containing engineering drawing

Description automatically generated

Using these results, we apply morphological edge detection with border finding:

A picture containing background pattern

Description automatically generated

Filling the outer border:

A picture containing text

Description automatically generated

Delete everything connected to the outer parameter:

A picture containing text

Description automatically generated

Apply regional filling:

A picture containing text, dark

Description automatically generated

Remove unwanted leftovers:

A picture containing text, light, dark

Description automatically generated

Label:

A picture containing text, dark, light, image

Description automatically generated

From here we are easily able to extract all the bounding boxes as well as the contours of each fragment

Diagram

Description automatically generated

After that we remove all the borders that Length/Hight is more than 5 and Hight/Length is more than 4.5. by that we remove the unwanted regions Qr code

Description automatically generated with medium confidence

A picture containing text

Description automatically generatedDiagram

Description automatically generatedA picture containing text

Description automatically generatedQr code

Description automatically generated with medium confidenceResults for all the images:

A picture containing text

Description automatically generatedText

Description automatically generated

Things that we have tried:

* Smoothing
* Morphological operation
* Adaptive threshold

Here we got a problem because we have a different contrast of the small pieces and the background.