

Automatic Detection and Recognition of Children Toy Cubes

By Hussain Ghonem

14th of June 2021

Table of Contents

1. Contents of the Project

2. The Distribution of cases

- a) **Project1.m**
- b) **Project2.m**
- c) **Project3.m**
- d) **Project4.m**
- e) **Project5.m**
- f) **Project6.m**
- g) **Project7.m**

3. Project Assigned

- a) **Main Idea**

4. Methods used in detecting Words in Toy Cubes

- a) **detectMSERFeatures**
- b) **OCR**
- c) **Morphological operations**

5. Cases Solved

- a) **Case 1**
- b) **Case 2**
- c) **Case 3**
- d) **Case 4**
- e) **Case 5**

f) Case 6

g) Case 7

h) Case 8

Contents of the Project:

- TestCases Folder
- Hussain170440_ToyCubeWordReader.docx
- Hussain170440_ToyCubeWordReader.pdf
- Project1.m
- Project2.m
- Project3.m
- Project4.m
- Project5.m
- Project6.m
- Project7.m

The Distribution of cases:

Project1.m:

Case 1:

1.1 & 1.3

Case 3:

3.2

Case 4:

4.2(6,4,I, and 3 only)

Case 5 Tilt:

5.1(B and Y only)

Case 6:

6.4 ,6.3(A,C only), 6.2, and 6.1(X,H, and 6 only)

Case 7 TightCubes:

7.1 and 7.3(3,2, and 5 only)

Case 8:

8.9, 8.C, 8.E, and 8.O

Project2.m:

Case 1.2

Project3.m:

Case 2.1

Project4.m:

Case 3.1

Case 8:

8.M

Project5.m:

Case 3.3

Project6.m:

Case 4.1

Project7.m:

Case 8.Q

Main Idea

Children cubes are one of the most common toys to teach numbers, letters and common words, as shown in Figure 1. Usually, parents are required to sit with their children many times to help them during the learning process. Automating this process will help children learn without continuous supervision from their parents, and will make parents happy ☺. It could also help people with general learning disability.

Example:



Figure 1: Case3/3.3.jpg

This application aims to automatically detect and recognize the shape on one of the cube sides. As a start, assume that the camera is facing one of the cube surfaces with small tilting, and at different distances. The cube surface comes with different colors. It can be captured with any in-plane rotation angle on simple backgrounds. There are 10 initial shapes under consideration. The application should be able to automatically detect and recognize any of the 10 shapes on one or more cubes from the given image

Example:

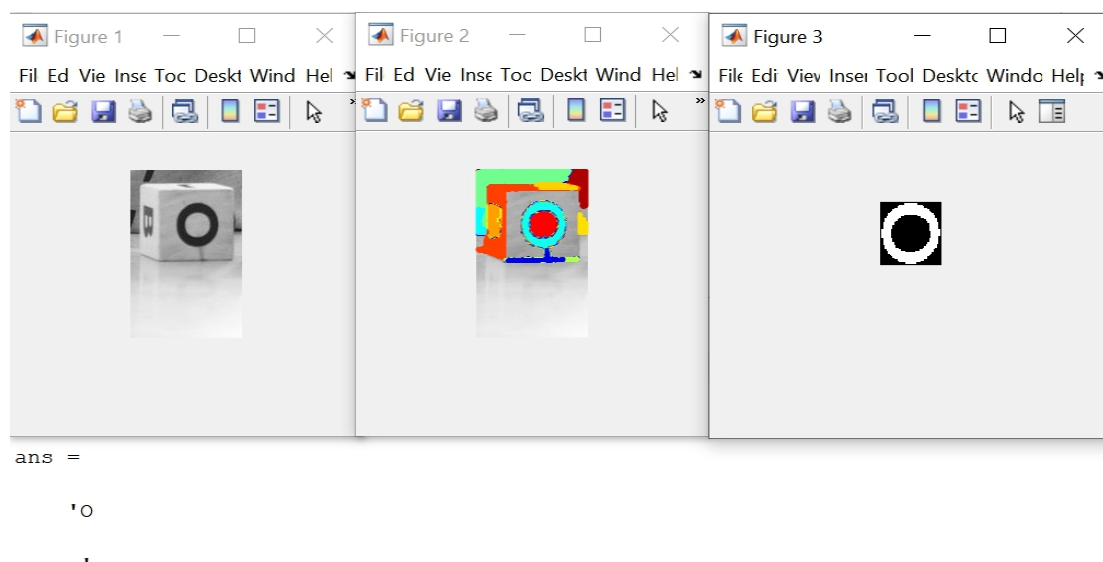


Figure 2: Case1/1.1.jpg Solution

Methods used in detecting Words in Toy Cubes

1. detectMSERFeatures
2. ocr
3. Morphological operations.

1. detectMSERFeatures:

detectMSERFeatures is a function that returns an MSERRegion object which is a region object. This function takes the image grayscale as input then uses the Maximally Stable External Regions algorithms to find regions. The MSER algorithm finds regions by using Intensity Threshold Levels ranging it by using the ThresholdDelta parameter. The smaller the ThresholdDelta parameter the more implementation for the ranging for the Intensity Threshold Levels takes. The bigger the ThresholdDelta parameter the less implementation for the ranging for the Intensity Threshold Levels takes.

I used the detectMSERFeatures function to detect the words in the cube since the word in the cube would have a different colour than the cube which makes the word region a Maximally Stable External Region. Then I used the cc Connected Component structure which is a structure with four fields that's used for measuring the MSER's region properties using the regionprops function. As we can see in Figure 3 the word 'O' in the cube is coloured with a different colour than the surrounding objects around. That's happened because the detectMSERFeatures have successfully defined the word 'O' as an independent object which helped in reading the word. I have used a for loop to read all the objects in the image.

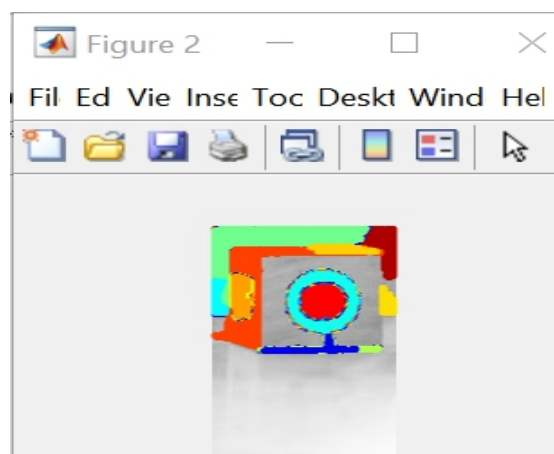


Figure 3: Case1/1.1.jpg MSERRegions

2. OCR:

The OCR means Optical Character Recognition which is a function that I have used to recognize the words in the images of the cubes. The OCR takes the image as input, then recognize the words in the image and print it out when calling OCRresults.Text where OCRresults is equal to the OCR that has received image as an input.

The OCR function can take more inputs beside the image to enhance its abilities in detecting characters in images for example adding a CharacterSet will help the function to increase its accuracy in detecting characters. I have added CharacterSet as another input besides the image. In the CharacterSet I have entered all the numbers and the letters to help me in detecting the characters in the image. The OCR contains another helpful parameter which is WordConfidences. That can be called by OCRresults.WordConfidences the WordConfidences parameter is a ratio from 0 to 1 where 0 means that the OCR function doesn't think that the current object is related to any character in the CharacterSet and 1 means the OCR function strongly thinks that the current object is related to any character in the CharacterSet. I have used the WordConfidences parameter to see if the objects are related to any character in the character list then set it to only accept an object that has more than 0.8 in WordConfidences which means I only read objects that the OCR thinks there's more than 80% similarity to a character in the character list.

3. Morphological operations:

The Morphological operations are a set of image processing operations that process pictures based on their shapes. In morphological operations, each pixel is corresponding to the value of the other neighbourhood pixels. The imdilate adds pixels to the boundaries of the objects the exact number of pixels are corresponded based on the neighbourhood of the actual pixels. The imerode removes pixels from the boundaries of the objects the exact number of pixels are corresponded based on the neighbourhood of the actual pixels.

I have used imdilate to improve the boundaries in some images that are lacked sharpening in its word's boundaries to the cube. Then used imerode to clean the inside of the word from scratches and other issues.

Cases Solved:

Case 1:

1.1, 1.2, and 1.3

Case 2:

2.1

Case 3:

3.1, 3.2, and 3.3

Case 4:

4.1 and 4.2(6,4,I, and 3 only)

Bonus:

Case 5:

5.1(B and Y only)

Case 6:

6.4 ,6.3(A,C only), 6.2, and 6.1(X,H, and 6 only)

Case 7:

7.1 and 7.3(3,2, and 5 only)

Case 8:

9, C, E, M, O, and Q