Image Processing Project

StudentID (Cand.No): 244848
Author: Joakim Martin Torsvik

Date: March 31th, 2022

Introduction

The following report shows the results of the image processing project. Here I will explain what I did to make my model and the challenges I faced on my way to making it. Below the report is a table of all the images from sample of colours with no borders. Because I managed to find all the colours in all the images, I don't have any notes on specific images other than if the outputs are flipped or not.

When I called on the images, I called on them directly from the local disk folder where I stored the images, so in my model you will see each one. To change the image, I just changed the part of the string.

Functions

LoadImage function: This function was fairly straight forward. The only hinderance was that some of the images was sized differently, so I resized every image into 480x480.

FindCircles function: To isolate the circles I used thresholding to reduce the intensity of the bright points and increased the intensity of the dark points so that the circles are isolated in the image. I then enlarged the circles using imdilate, removed further noise with a median filter, eroded unwanted spots and filtered them out, then enhanced the circles again. Here I needed to tweak the threshold to remove all the noise on the images, but still be able to find the dark circles. Especially org_1 and org_2 was difficult to fit in with the rest of the images.

Then I calculated the centroids and the major- and minor axis and the orientation of the regions and used a Hough transformation to transform the circles seeing as some of the circles in the projection images are stretched and doesn't look like circles.

CorrectImage function: From the previous function I used the coordinates to warp the image into a quadratic square. For the projection images it meant that I had to project the images whilst trying to maintain the shape of the colours. Sometimes it wasn't completely successful because the images were warped too much (see output in image proj1_3 for reference).

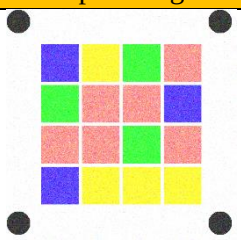
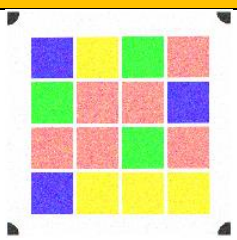
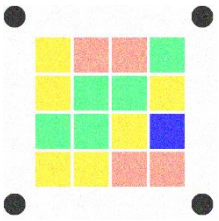
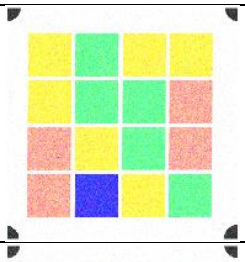
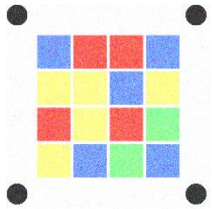
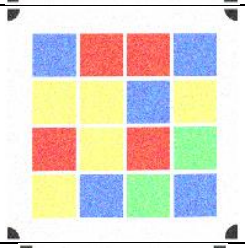
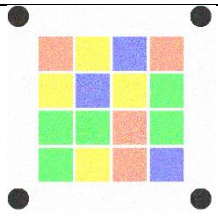
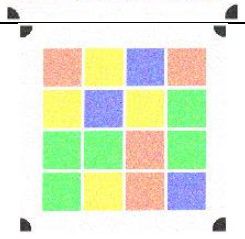
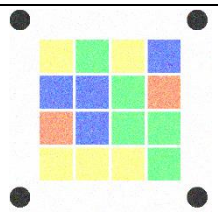
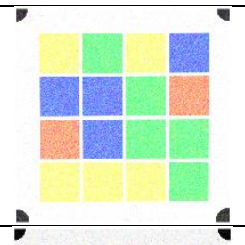
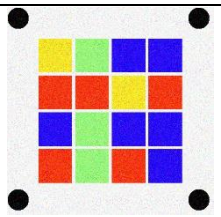
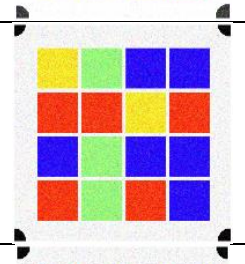
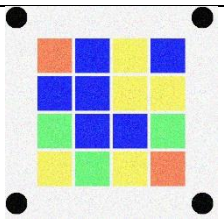
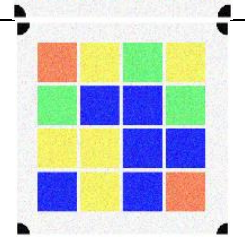
FindColors function: With the now corrected image from the previous function I converted the colour space into LAB and applied a mean filter to remove the noise from the colours so when I would take a sample from the squares, I could get the average colour from the squares. Because the colour points are located in a symmetric square, I needed only to write a 4D vector which I used to locate the LAB colour space in the square. Then I defined each the colour of each letter (r, g, b, y) to the LAB-scale and calculated the minimum distance from

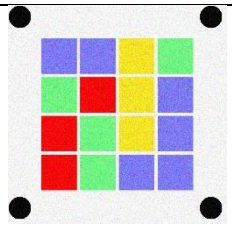
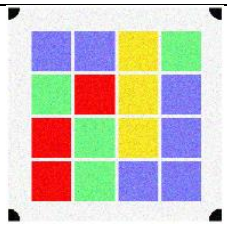
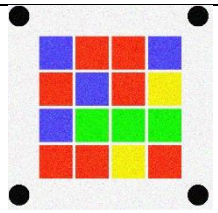
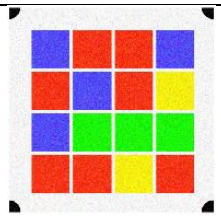
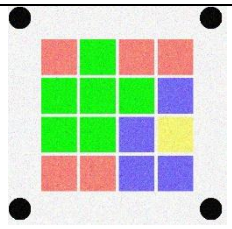
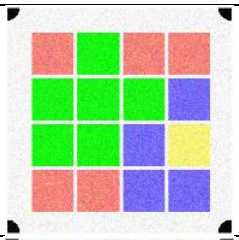
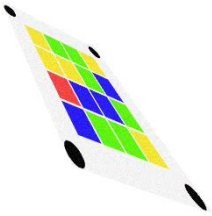
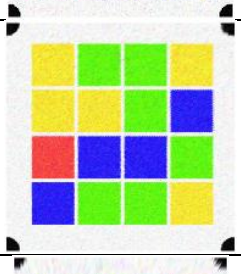
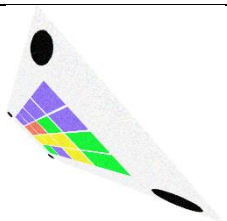
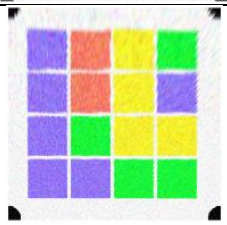
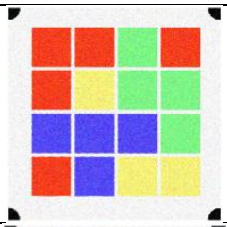
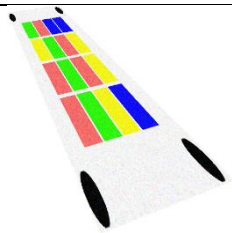
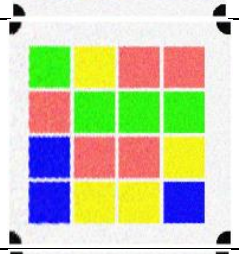
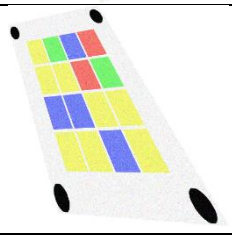
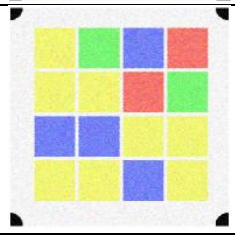
the observed colours to the defined colours I just mentioned, which finally gave me the matrix.

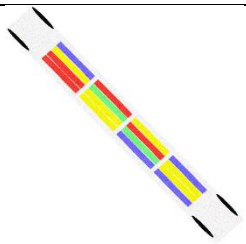
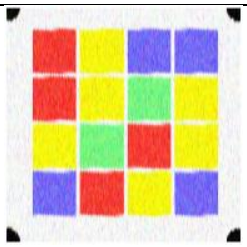
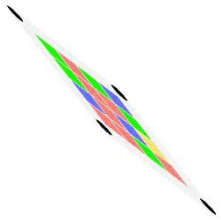
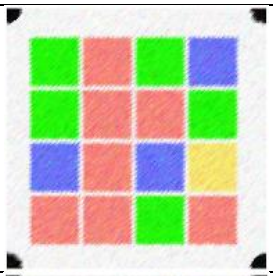
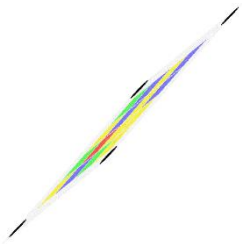
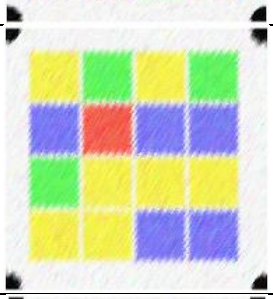
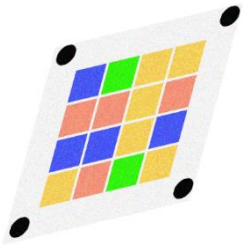
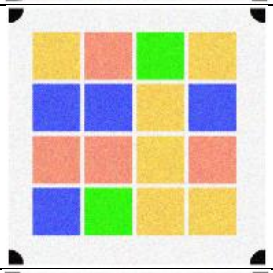
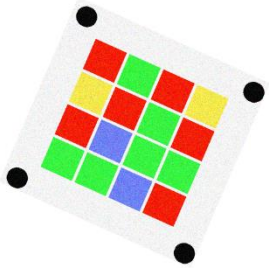
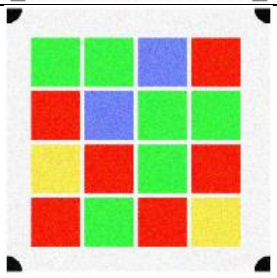
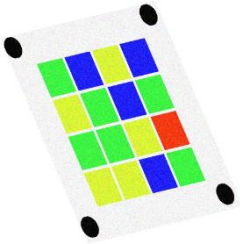
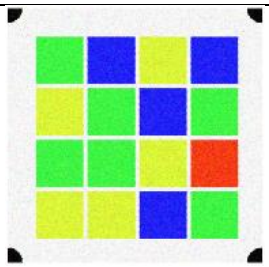
MainFunciton function: In the main function I simply called upon the above-mentioned functions and displayed the results in a 4x4 matrix.

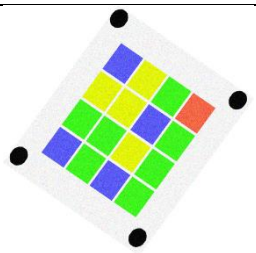
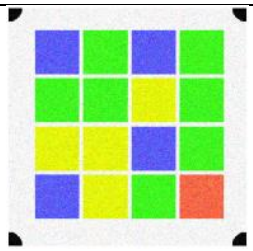
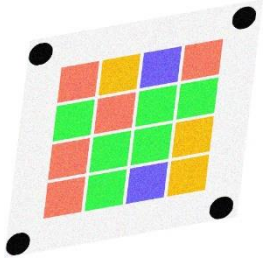
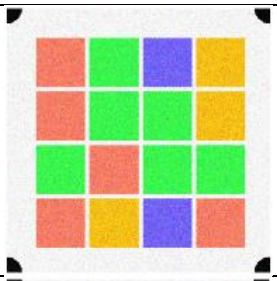
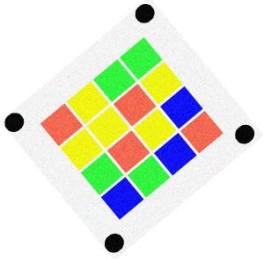
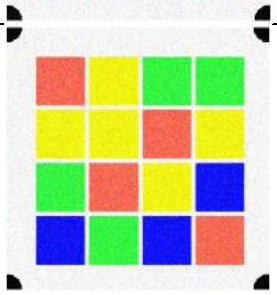
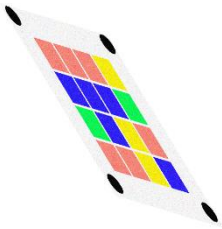
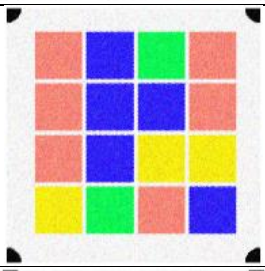
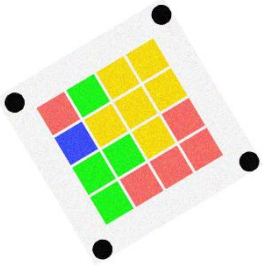
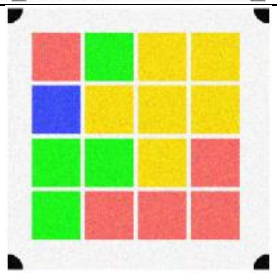
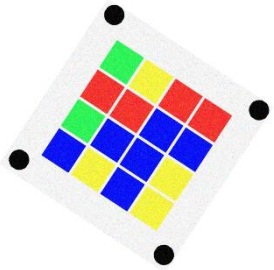
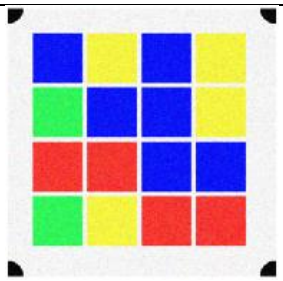
Photographs

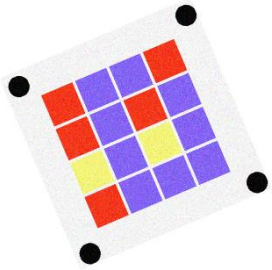
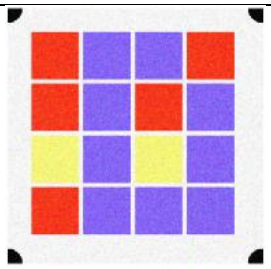
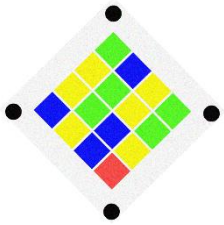
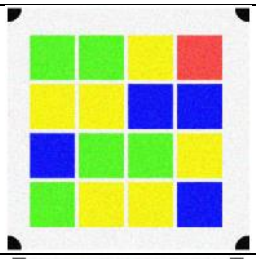
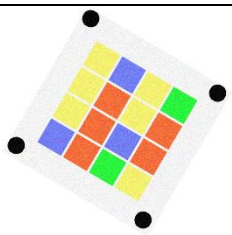
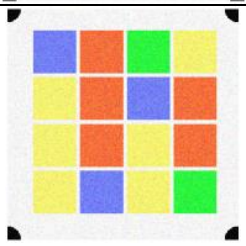
The assignment came with a selection of pictures where the colour squares have been photographed in different situations. Every photo contains two black rectangles with the same shade as the black circles in the squares. With too much smoothening such as what I did in the FindCircle function (with imdilate, imerode and medfilt2 functions) the squares might become a big circle or several small circles that will distort the coordinates of the corner squares. The light intensities could also play a part in identifying the colours of the images. Images 32 and 35 contain shadows that break in the middle of the photos, and image 34 have a bright glare in the square, all which would distort the intensity of the images.
Image 38 is placed on a whiteboard with many other shapes and distracting background shapes that can disrupt the model.

Index	Filename	Input Image	Output Image	Output	Success	Notes
1	noise_1			$\begin{pmatrix} b & y & g & r \\ g & r & r & b \\ r & r & g & r \\ b & y & y & y \end{pmatrix}$	Yes	No notes
2	noise_2			$\begin{pmatrix} y & g & y & y \\ y & g & g & r \\ r & y & g & r \\ r & b & y & g \end{pmatrix}$	Yes	Output is flipped
3	noise_3			$\begin{pmatrix} b & r & r & b \\ y & y & b & y \\ r & y & r & y \\ y & b & y & b \end{pmatrix}$	Yes	No notes
4	noise_4			$\begin{pmatrix} r & y & b & r \\ y & b & y & g \\ g & g & r & g \\ g & y & r & b \end{pmatrix}$	Yes	No notes
5	noise_5			$\begin{pmatrix} y & g & y & b \\ b & b & g & r \\ r & b & g & g \\ y & y & y & g \end{pmatrix}$	Yes	No notes
6	org_1			$\begin{pmatrix} y & g & b & b \\ r & r & y & r \\ b & g & b & b \\ r & y & r & b \end{pmatrix}$	Yes	No notes
7	org_2			$\begin{pmatrix} r & b & y & b \\ b & b & y & y \\ g & b & b & g \\ y & g & y & r \end{pmatrix}$	Yes	Output is flipped

8	org_3			$\begin{pmatrix} b & b & y & g \\ g & r & y & b \\ r & g & y & b \\ r & g & b & b \end{pmatrix}$	Yes	No notes
9	org_4			$\begin{pmatrix} b & r & r & b \\ r & b & r & y \\ b & g & g & g \\ r & r & y & r \end{pmatrix}$	Yes	No notes
10	org_5			$\begin{pmatrix} r & g & r & r \\ g & g & g & b \\ g & g & b & y \\ r & r & b & b \end{pmatrix}$	Yes	No notes
11	proj_1			$\begin{pmatrix} y & g & g & y \\ y & y & g & b \\ r & b & b & g \\ b & g & g & y \end{pmatrix}$	Yes	No notes
12	proj_2			$\begin{pmatrix} b & r & y & g \\ b & r & y & b \\ b & g & y & y \\ b & b & g & g \end{pmatrix}$	Yes	Output is flipped
13	proj_3			$\begin{pmatrix} r & r & g & r \\ r & y & g & g \\ b & b & b & g \\ r & b & y & y \end{pmatrix}$	Yes	No notes
14	proj_4			$\begin{pmatrix} g & y & r & r \\ r & g & g & g \\ b & r & r & y \\ b & y & y & b \end{pmatrix}$	Yes	Output is flipped
15	proj_5			$\begin{pmatrix} y & g & b & r \\ y & y & r & g \\ b & b & y & y \\ y & y & b & y \end{pmatrix}$	Yes	No notes

16	proj1_1			$\begin{pmatrix} r & y & b & b \\ r & y & g & y \\ y & g & r & y \\ b & r & y & b \end{pmatrix}$	Yes	Output is flipped
17	proj1_2			$\begin{pmatrix} g & r & g & b \\ g & r & r & g \\ b & r & b & y \\ r & r & g & r \end{pmatrix}$	Yes	No notes
18	proj1_3			$\begin{pmatrix} y & g & y & g \\ b & r & b & b \\ g & y & y & y \\ y & y & b & b \end{pmatrix}$	Yes	No notes
19	proj1_4			$\begin{pmatrix} y & r & g & y \\ b & b & y & b \\ r & r & y & r \\ b & g & y & y \end{pmatrix}$	Yes	Output is flipped
20	proj1_5			$\begin{pmatrix} g & g & b & r \\ r & b & g & g \\ y & r & g & r \\ r & g & r & y \end{pmatrix}$	Yes	Output is flipped
21	proj2_1			$\begin{pmatrix} g & b & y & b \\ y & g & b & g \\ g & g & y & r \\ y & y & b & g \end{pmatrix}$	Yes	No notes

22	proj2_2			$\begin{pmatrix} b & g & b & g \\ g & g & y & g \\ y & y & b & g \\ b & y & g & r \end{pmatrix}$	Yes	Output is flipped
23	proj2_3			$\begin{pmatrix} r & g & b & y \\ r & g & g & y \\ g & r & g & g \\ r & y & b & r \end{pmatrix}$	Yes	Output is flipped
24	proj2_4			$\begin{pmatrix} r & y & g & g \\ y & y & r & y \\ g & r & y & b \\ b & g & b & r \end{pmatrix}$	Yes	No notes
25	proj2_5			$\begin{pmatrix} r & b & g & r \\ r & b & b & r \\ r & b & y & y \\ y & g & r & b \end{pmatrix}$	Yes	Output is flipped
26	rot_1			$\begin{pmatrix} r & g & y & y \\ b & y & y & y \\ g & g & y & r \\ g & r & r & r \end{pmatrix}$	Yes	No notes
27	rot_2			$\begin{pmatrix} b & y & b & y \\ g & b & b & y \\ r & r & b & b \\ g & y & r & r \end{pmatrix}$	Yes	Output is flipped

28	rot_3			$\begin{pmatrix} r & b & b & r \\ r & b & r & b \\ y & b & y & b \\ r & b & b & b \end{pmatrix}$	Yes	No notes
29	rot_4			$\begin{pmatrix} g & g & y & r \\ y & y & b & b \\ b & g & g & y \\ g & y & y & b \end{pmatrix}$	Yes	Output is flipped
30	rot_5			$\begin{pmatrix} b & r & g & y \\ y & r & b & r \\ y & r & y & r \\ y & b & y & g \end{pmatrix}$	Yes	Output is flipped