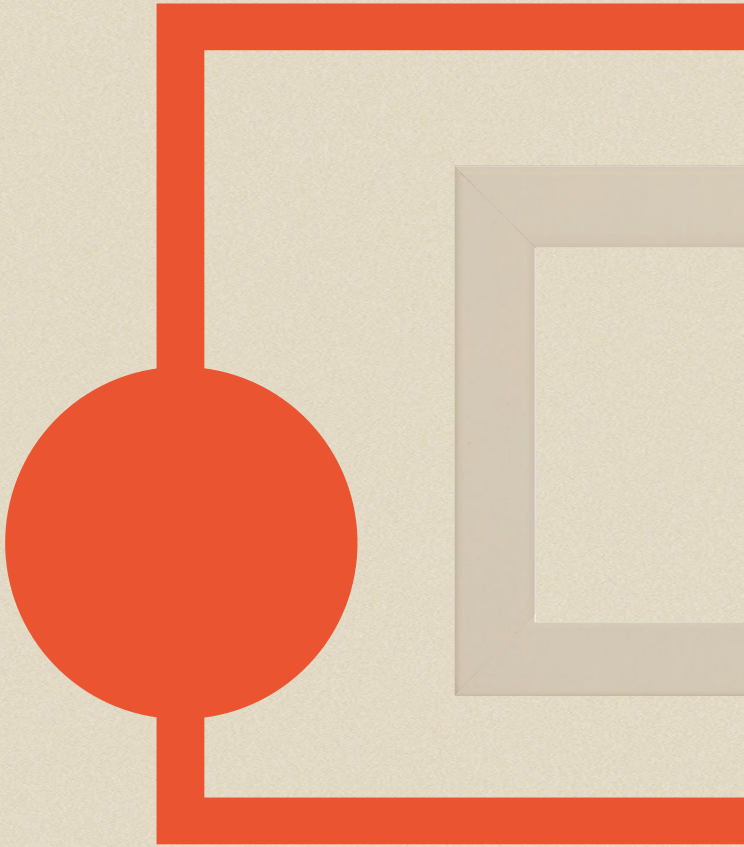
Time Reader

Analogue to Digital Conversion Application

18BCE2098 - Satvarsh Gondala



01

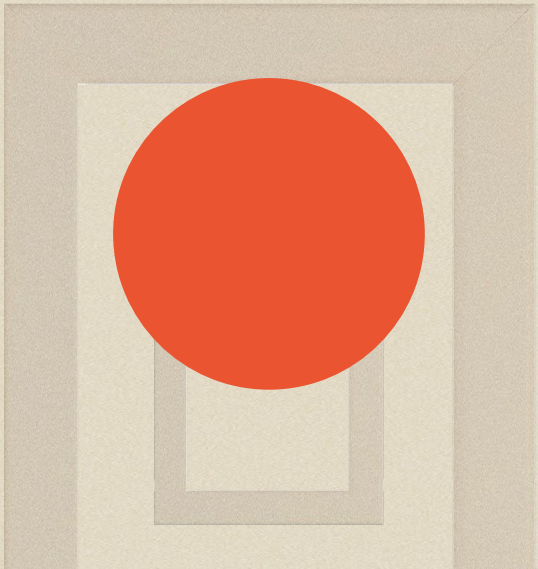
FOUNDATION

Abstract, Intro, Reason and Application

02

METHODOLOGY

Working, Dataset, Process, Improvements



01

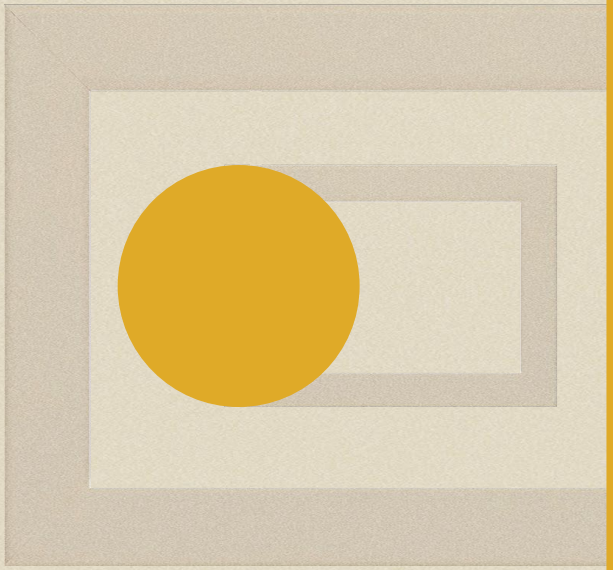
Foundation

Abstract, Intro, Reason and Application

See **report** for the Reason and Application of this project

Abstract

We already know the importance of Image Processing and how it contributes in opening up so many streams in the field of Computer Science. The main forward for this project is for us to make an application toolkit which can use multiple algorithms to successfully convert an analogue clock to digital one and read the data within it. The main object here being a clock is a variable, but the concepts we learn will always be constant hence, the objective is to learn as much about image processing as possible.



Why This Topic?

Modular

This project is very customizable with many branches

Understanding

To learn as much about Image Processing as possible

GSOC

All three of us plan to enroll to GSOC for coming summer for a project that is similar

Scope

This topic can be easily expanded as needed for coming reviews

See **report** for complete Introduction

Introduction

While, the Idea to combine a clock and image processing might seem like two unrelated things, it is quite the opposite and not unconventional as it sounds. Image Processing has been used in multiple use cases from identifying and detecting cancer to readily segregating and analyzing multiple images in parallel. The clock while is a variable is a very strong starting source as it can come in various shapes and forms, while also having a common image that defines an analogue clock. Given how there are more than thousands of variations when it comes to clock, the complexity of the project can really exceed than most of the projects in theory.. This really sums up one of the factors at which this project excels, tailoring and customization ; with so many limitless options to choose from on top of which you have an algorithm you have so many variables and so many concepts that needs learning, exactly serving our main goal which is not only to provide an application toolkit which you can customize and incorporate into your own projects, but also to learn as much about image processing as possible.

Does this have an application?

Yes, the said project does not have to be restricted to just Clocks, the goal at the end is to take the concept and expand it to many variables like speed-o-metre, gauge and other interfaces. Even if we give an application with respect to clocks, with the ability to convert any clock image into a numerical value, will give us a way to analyze 100s of images instantly in the event of a crime (example) where you have to know a particular time frame in a room where there are clocks. This can be done by identifying the clock and segregating all the images into specific timezones. While this might sound trivial in concept, the possibilities are limitless as our aim is to provide an application that can be tailored to your preference.

See **report** for Literature Survey, base research paper

02

Methodology

Working, Dataset, Process, Improvements

Working

Our project is made up of three variables

Input

- Real-Life Image
- Computer Drawn Image

Process

- Sobel Edge Detection
- MVVM Pattern
- Hough Transform

Output

- Reading
- New Clock
- Simplified Clock

INPUT - Datasets Used

Computer-Drawn Images

contains 50K images of computer drawn analogue clocks

Real-Life Images

A zip file consisting of various real-life images of analogue clocks

This is only for **ONE** combination

Process (Sobel Edge Detection)

Input

Set an image variable

Type

Set a debug variable

Reading

Read the Image

Centre

Find the center of the clock using Sobel Edge Detection

Full Image

Get the clock drawing by applying hough line

Result

Determine angle between lines to get the value of the clock

REVIEW - II - Focus

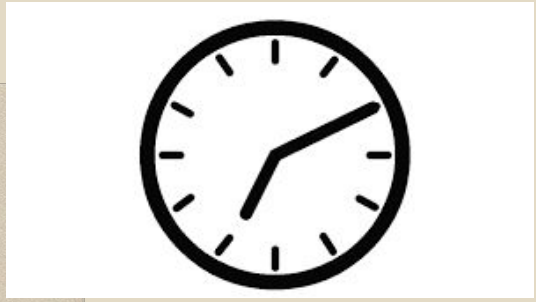
Input Variation

We accounted for Computer generated images, real life images, real life distorted image, real life angled images

Integration

Have everything run within a single program. We integrated different processes within matlab (migrated from jupyter notebook)

Input Types



Computer Generated

Angled Images

Real-life Images

Noised Images

Process - I

1

GRAYSCALING

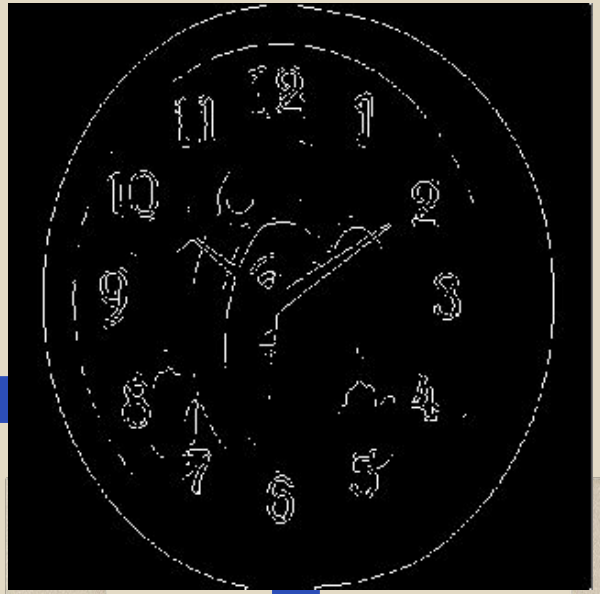
Every Image is converted to grayscale version to perform detection better

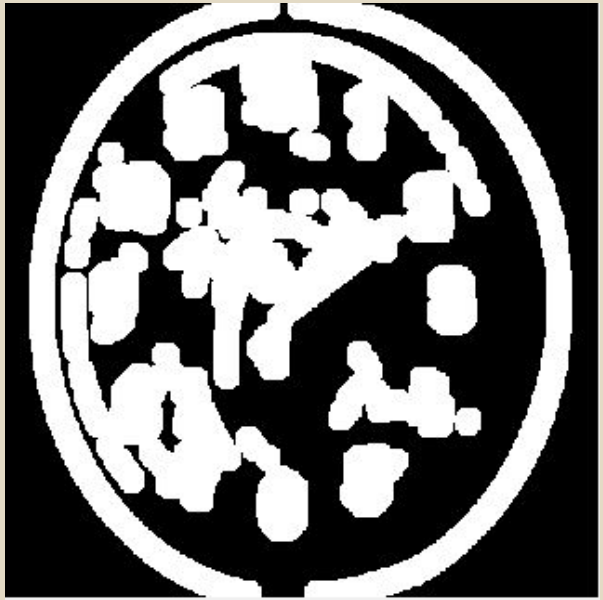
Process - I

2

Edge Detection

Using sobel operator we perform sobel edge detection and then use strel function





Process - I

3

Centre Calculation

Boundary and centre detection

Process - II

4

Hands Identification

Hough Lines Transform is used to detect the hours, minutes and seconds hand

Process - II

3 Time Calculation

Once hands are identified time is calculated

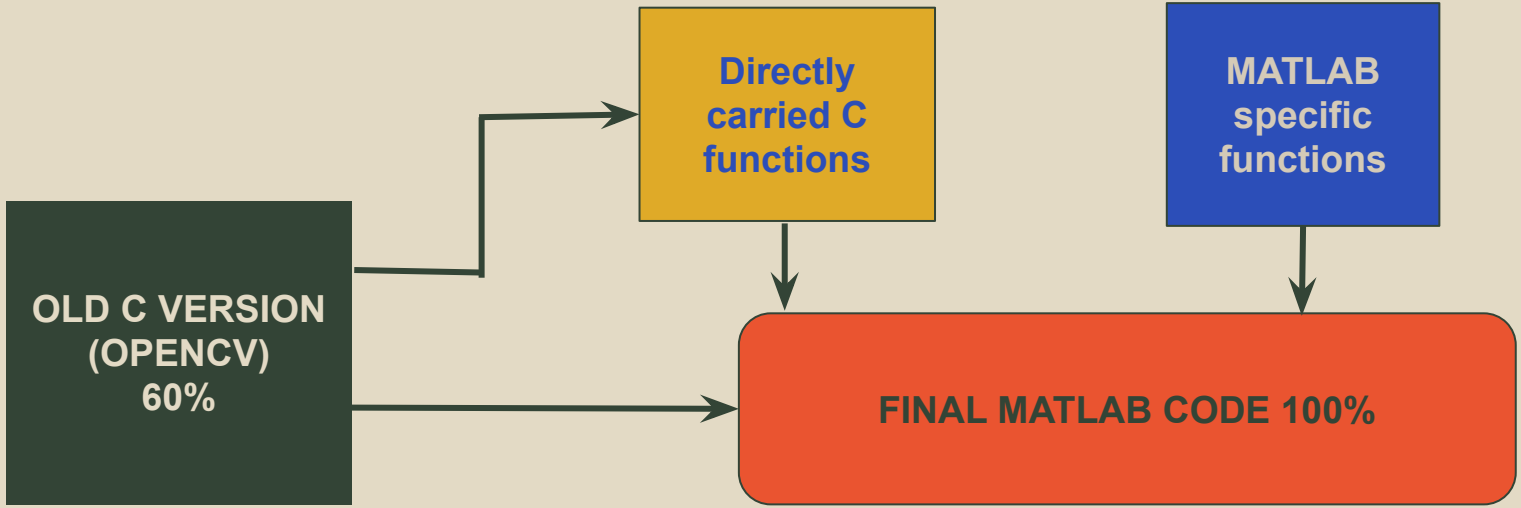
The Clock reads 10:9.7447
Image Processing - Review 2

I	450x450x3 uint8
imageurl	'D:\18BCE2098\c...
info	1x1 struct
j	1
longest	1x1 struct
max_len	0
maxxy	[450,450]
minute	9.7447
minxy	[0,0]
newlongest	1x1 struct
v2	[0,1,0]
vhour	[61.2500,39,0]
vminute	[-84.7500,52,0]

angle1	58.4680
angle2	302.4863
arrowslines	1x2 struct
center	[210.2500,222]
debug	1
edges	450x450 logical
finallines	1x2 struct
hour	10

LIVE DEMO

Formation of the code



Improvements/Changes for review - 3

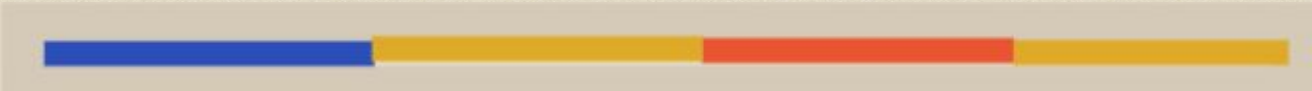
- Hough Lines Transform is completely redone, we added the ability to calculate new centre using struct function and now we determine the closest centre form that. Improved accuracy.
- The entire code is optimized with more defined loops, removed all unused variables, reused old variables, reduced code by 60 lines. Improved speed
- Tested and included two more data types. This code can measure border shapes more accurately and we explored clocks outside of circle and also tried different lighting conditions.
- Divided calculation algorithm within hough line and main code, leading to faster computation.

TESTING

We have a total of 120 Images of clocks divided into four types :

Computer Generated	30 Images
Real-life Images	30 Images
Angled Images	30 Images
Noise Images	30 Images

Final Ratio of all data types :

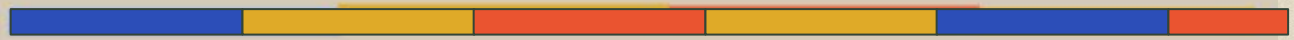


TESTING - II

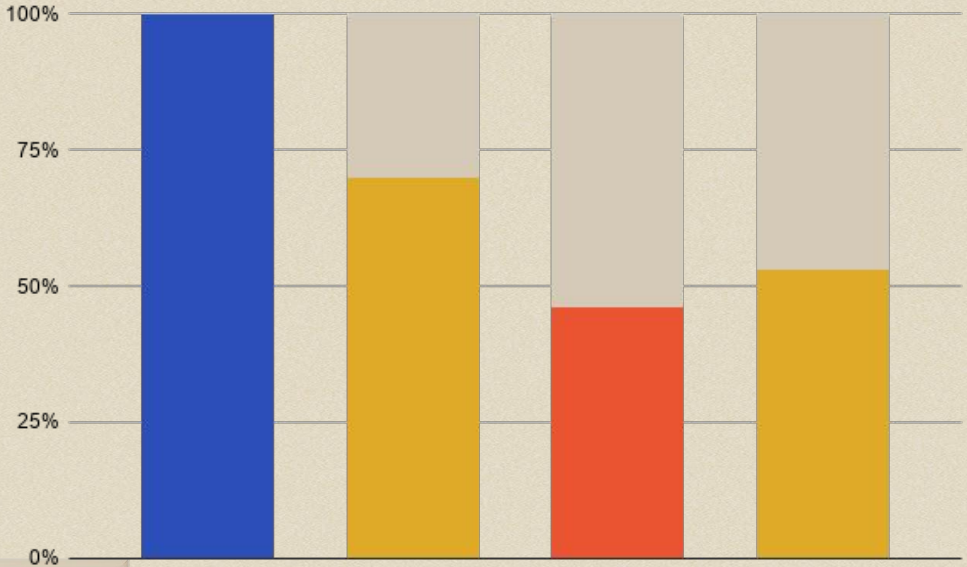
We have a total of 170 Images of clocks divided into four types :

Computer Generated	30 Images
Real-life Images	30 Images
Angled Images	30 Images
Noise Images	30 Images
Different Shapes	30 Images
Challenging Images	20 Images

Final Ratio of all data types :



RESULTS ANALYSIS



Computer Generated

30/30

100%

Real-Life

21/30

70%

Angled Images

14/30

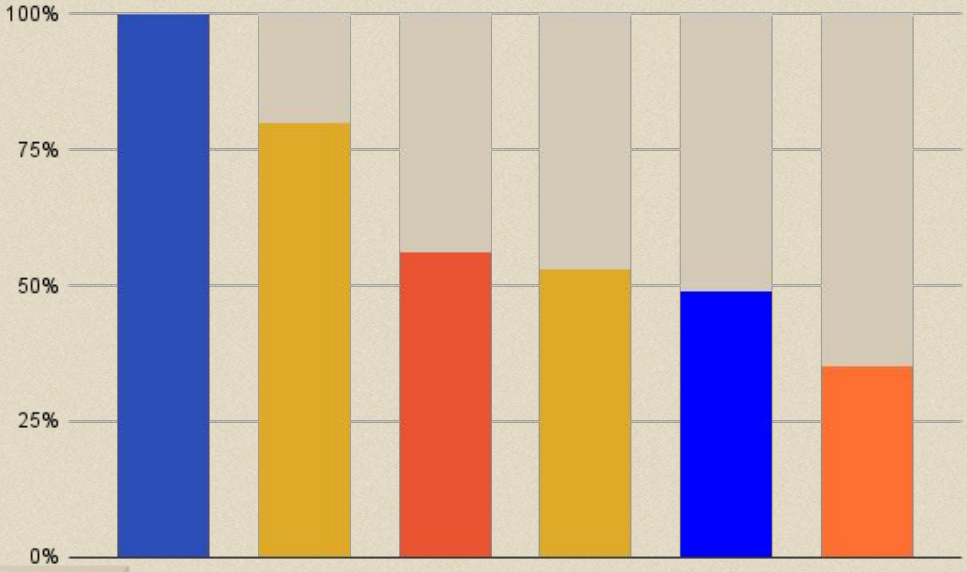
46%

Noise Images

16/30

53%

RESULTS ANALYSIS - II



Computer Generated

30/30

100%

Real-Life

24/30

80%

Angled Images

20/30

67%

Noise Images

17/30

56%

Different Shaped

16/30

49%

Challenging Images

7/20

35%

Improvements in Results

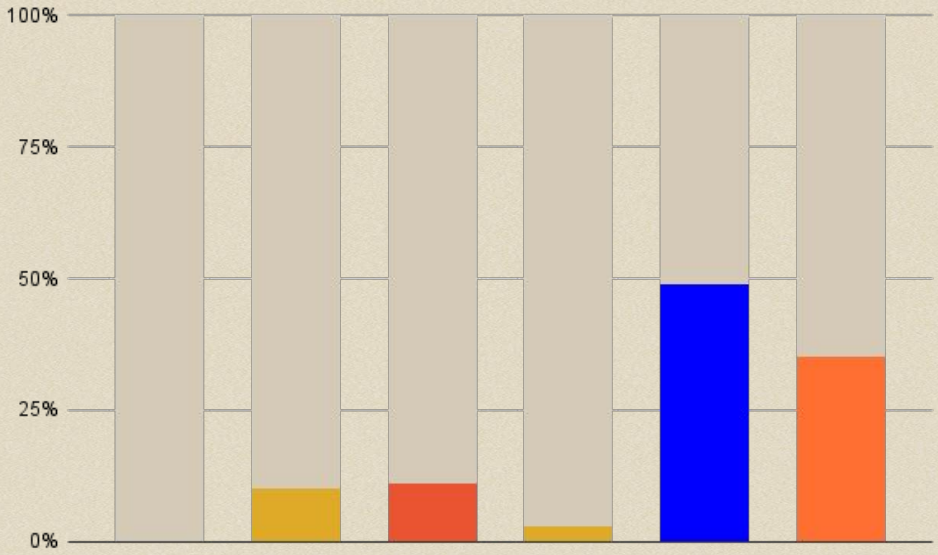


Image Type	Improvement for Review-3 (accuracy)
Computer Generated	+ 0%
Real-Life Images	+ 10%
Angled images	+ 11%
Noised Images	+ 3%
Different Shapes	+ 49%
Crafted/Challenging images	+ 35%

Plan for the coming reviews (review-3)

We exceed the goals for review 3 and did three additional tasks

	Review 1	Review 2	Review 3	Additional Goals
Have at least three processes	✗	✗ 1-2 processes	✓ 2-4 processes	✓ 6 processes
Have Implementation	✗	✓ For a select few	✓ All + Integration	✓ Additional optimization, improved speed and accuracy
Dataset Supported	✗	✓ Computer Generated	✗ ALL Datatypes + 1 additional device	✓ Two new data types

THANK YOU

Feel free to ask any questions