COMP 4102 Final Project: Analog Clock Reader

Majd Al Khany - 100977501
Layne Koftinow-Mikan - 101013563
Github page link: https://github.com/majdalkhany/analog-clock-reader

Summary

The purpose of this project is to develop a computer vision system which scans an image of an analog clock, determines the time represented on it, and returns a string representation of the time in hours, minutes, and seconds. This will be accomplished by using edge detection to detect the clock's hands and an OCR algorithm to detect the clock's numbers. The algorithm will be designed to support clocks which are missing key values such as a seconds hand or numbers. Additionally, the algorithm will support clocks which are rotated by and/or viewed at some unknown angle.

Background

Many programs already exist to read the time from an analog clock. The purpose of this project is to make a similar program which is much more verbose in the input it accepts. Ideally, this program would be implemented as a mobile application so the user is able to easily take a photo of a clock and input it to the program. However, given the time restrictions of this project, it will be developed in Python and tested on desktop using photographs we have taken ourselves and images pulled from the internet. If time permits, it can be converted into a mobile application.

The Challenge

Developing the analog clock reader will be challenging because it will be designed to be as verbose as possible. Provided only an input image, the program will output a string representation of the clock's time while respecting as many different cases as possible. This includes the ideal clock scenario that is correctly oriented and contains an hours, minutes, and seconds hand; a clock without a seconds hand; and a clock without numbers. Additionally, the reader will support a clock which is not oriented correctly (ie. it is rotated at some unknown angle) and a clock viewed from an angle (ie. not viewed from directly in front). In either of these cases, the clock would need to detect the misorientation and/or misalignment and account for it in its calculation of the time. It should be noted that, if a clock has no numbers, it must be assumed to be oriented correctly, otherwise there is no way to determine its orientation. In developing this project, we hope to gain a better understanding of edge detection, OCR (optical character recognition), and other algorithms which can detect and account for objects which are rotated or viewed from angles.

Ideal clock	Clock without seconds	Clock without numbers	Misoriented clock	Misaligned clock
				

Fig. 1 - Examples of input images which will be supported

Goals and Deliverables

The minimum of what we plan to achieve is the development of a program which is able to scan an image of a correctly oriented analog clock and determine the time represented on it. This should work whether or not the clock has numbers or a seconds hand. Afterwards, we will also attempt to program the ability to read a misoriented clock (ie. rotated at some unknown angle) as well as the ability to detect and account for the clock being misaligned (ie. captured from some unknown angle). If time permits, we will create a basic mobile application which runs the algorithm so as to be more user-friendly.

The project's success can be evaluated by comparing the actual time on the clock to the output of the algorithm. This would be done by simply recording the expected output for each image, comparing it to the actual output, and calculating the difference between them. This can be done automatically through test cases.

Neither of us are familiar with computer vision algorithms so part of the challenge thus has been determining a concept that was within the scope of this project. Based on what we have learned so far, alongside some additional outside research, this project seems to be realistic given the allotted time.

Schedule

Week	Days	Layne's Tasks	Majd's Tasks
1	Feb. 2 - Feb. 8	Research and learn required algorithms	Research and learn required algorithms
2	Feb. 9 - Feb. 15	Design algorithms and program structure	Design algorithms and program structure
3	Feb. 16 - Feb. 22	Implement edge detection of hands	Implement OCR reading of numbers
4	Feb. 23 - Feb. 29	Implement time calculation of ideal clock	Implement OCR reading of numbers
5	Mar. 1 - Mar. 7	Account for rotated clock	Account for rotated clock
6	Mar. 8 - Mar. 14	Account for rotated clock	Account for rotated clock
7	Mar. 15 - Mar. 21	Account for misaligned clock	Account for misaligned clock
8	Mar. 22 - Mar. 28	Account for misaligned clock	Account for misaligned clock
9	Mar. 29 - Apr. 4	Develop mobile app if time permits	Improve existing algorithm implementations
10	Apr. 5 - Apr. 10 (due date)	Develop mobile app if time permits	Final testing before due date