MEE / MEEE

Visão Computacional / Computer Vision

Assignment 1 – Analog to Digital Time Conversion

1. Instructions

a) **Integrity and collaboration:** Students are encouraged to work in groups, so include the names of your collaborators in your write-up. Code should **NOT** be shared or copied among groups. **Plagiarism is strongly prohibited and may lead to the failure of this course.**

b) **Start early!** Especially those not familiar with Python and OpenCV.

c) **Submission:** Your submission for this assignment should be a zip file, **<CVis_2425_Assign1_JohnDoe_JaneDoe.zip>**, composed of your write-up, your Python implementations (including any helper functions), and results. Please make sure to remove any other temporary files you generated. Substitute the folder name and the filenames "JohnDoe" and "JaneDoe" by each group element's names and surnames.

Your final upload should have the files arranged in this layout:

<CVis_2425_Assign1_JohnDoe_JaneDoe.zip>

- < JohnDoe_JaneDoe_ Assignment1>.pdf => this is your report.
- **data** => location of the images that you are using.
- **papers** => folder to add papers with methods applied in the assignment.
- **code_python** => location of the scripts that you will develop. It would be best to explicitly mention **how to run** each file in a **readme.txt** file or similar.

d) **File paths:** Please ensure that any file paths you use are relative and not absolute. Avoid using complete paths like in the following example: cv2.imread(**' C:/name/Documents/assign1/data/xyz.jpg'**). Use relative paths like cv2.imread(**'../data/xyz.jpg'**) instead.

2. Implementation

This assignment uses as a reference image the **clock.png**, which displays an analogic time display clock. Transforming analog time into a digital format is fundamental in bridging the gap between human-readable information and machine interpretation, a core aspect of computer vision-based activities, as presented in Figure 1.

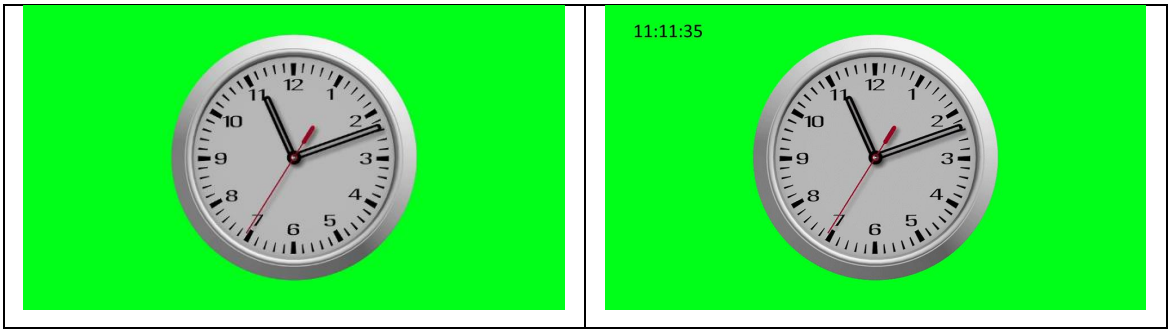

Figure 1: Examples of the reference image (left) and expected output (right).

In this task, students must extract visual features (the positions and angles of clock hands) and convert them into a digital format that a computer can process. The students are challenged with various tasks, such as segmenting the region defined by the clock in the image, segmenting the clock hands, extracting or considering its length, and estimating the corresponding hour/minute/second according to its positions. For each task, students will have to choose, justify, and document **the rationale and the processes** to obtain the expected result.

Every script and function that you write or use in this section should be included in the **code_python** folder, and you should **explicitly mention how to run it**. Please include the resulting images of the entire process in your report. Also, suppose the techniques/methods used don't belong to OpenCV's native set functions. In that case, students must document their sources (add the paper pdf file in the **papers** folder), indicating in your report the relevance of their use in this context.

2.1 Extraction of the analog time by applying grayscale conversion

In this task, it is proposed to use a greyscale conversion of the provided colored image to extract the face clock, which is the region of interest (ROI) of the provided image. From this step, the students may proceed to obtain the clock hand's position to extract their locations and infer the corresponding information (hour/minute/second). Finally, the result should be imposed over the original image after determining the time, overlaying the digital time.

For this task, students may use the techniques considered suitable for the task, including image transformations, filtering, thresholding, histogram analysis, etc.

2.2 Extraction of the analog time by applying color space conversion(s)

Similar to the previous task, the objective is to extract the analog time information. Still, in this case, the students must use approaches that use one or many different color spaces. Similarly, the students may use the techniques considered suitable for the task, as mentioned previously.

2.3 Run the algorithm in a video

Gather your previous methods and run them on a video (**clock.mp4**, also available in the data folder) to extract all the time changes frame by frame. The output of your algorithm should be a plain Comma Separated Value (CSV) text file (using the ".csv" extension) with the digital time (HH:MM:SS). The first file line must show your group number and corresponding student names and numbers, and the second line must show the video's frame and the obtained digital time, as exemplified below.

```
# Group X, <num1>, <name1>, <num2>, <name2>

Frame 1, 10:22:32

Frame 2, 10:22:33

Frame 3, 10:22:34
```

3. Report manuscript

For each previously mentioned task, ensure they run correctly and generate intermediate output images. Include the set of intermediate outputs in your report and **be critical** of the results you obtain. As examples:

- Did the code work well on all the images with a single set of parameters?
- How did the optimal set of parameters vary with images?
- Which step of the algorithm causes the most problems?
- Did you find any changes you could make to the algorithm to improve performance?

In your report, you should describe how well the code worked, the parameters' effect, and any improvements that can make it work better.

Table 1: Classification weights

Activity 2.1	Activity 2.2	Activity 2.3
40%	30%	30%