Practical work

Summary

This work aims to give students the opportunity to apply some of the image processing and analysis concepts covered in classes. Students must develop a program in C or C++, which operates on the video made available for this practical work, in order to obtain the desired set of information (see below). This work will culminate in the delivery of the developed source code, in addition to an oral defense of the work carried out.

Carrying out practical work

“VC-TP.zip” file presents the respective work description and objectives, also containing:

- Video “video_resistors.mp4” to be analyzed;
- Source code with example of reading/capturing and displaying video using the OpenCV library.

The work must be carried out in groups, with **groups consisting of 3 elements**.

Students are expected to use the knowledge acquired throughout Computer Vision classes and develop the functionalities necessary to achieve the objectives.

Note that all code must be written in C or C++ language, and the student can use (in addition to the OpenCV functions identified in the “CodigoExemplo.cpp” file, or similar functions, depending on the version of OpenCV) up to 2 more functions from the **OpenCV library**.

The use of external image processing libraries not covered in classes is not permitted; code made available in repositories (such as GitHub); as well as any other code whose authorship is not from the elements that make up the group.

The work must also be presented/defended by all members of the group in an **oral presentation/defense**, with **questions about the work being asked** to the various members of the group. It is not necessary to make any report or PowerPoint presentation. The presentation/defense should focus on the followed strategy and its implementation (source code), the difficulties experienced, and the resolution found for them. Students are expected to master all the source code used by the group to solve the practical work.

Goal

This work aims to develop an application that, by applying a set of algorithms, allows the counting and identification of electrical resistances present in a video. This application should be able to segment the different resistances (and only the resistances), identifying their value (expressing it in ohms) and relative location (which region of the image is occupied by each resistance) in each *frame* of the video.

The video shows: 1x 5600Ω resistor; 1x 220Ω resistor; 2x 1000Ω resistors; 1x 2200Ω resistor; and 1x 10000Ω resistor.

To identify the value of each resistance, use the following Resistance Table:

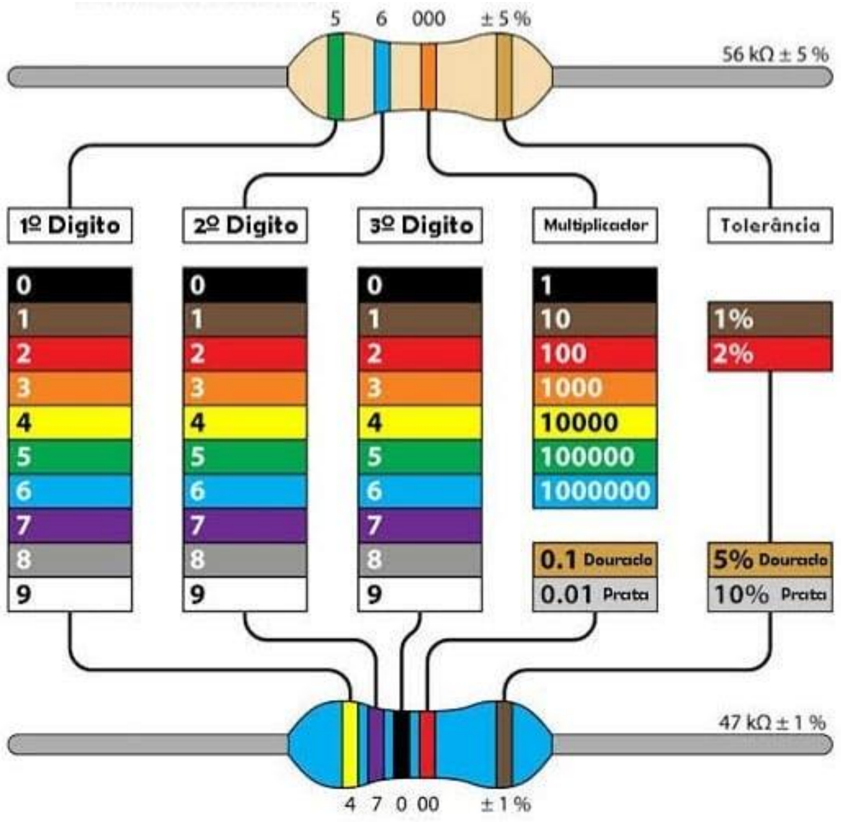

Figure 1– Resistance Table for 4 and 5 bands resistances.

The application must correctly detect each resistance present in each video frame, drawing the bounding box and center of mass (centroid) of the resistance in the video image. It must also present information on the video image, in text format, identifying the resistance value (in ohms). Video capture and display must be performed using functions from the OpenCV library. Image processing and analysis must be carried out through functions developed by the group.

Groups can consider that all resistances have a tolerance of ±5%.

System Specification

The computer vision system must include:

- Implementation of segmentation techniques (by color and/or brightness);
- Implementation of techniques for image enhancement (e.g., noise removal);
- Implementation of image analysis techniques that allow determining:
    - Area;
    - Bounding box;
    - Center of mass (centroid); among other information that the group considers relevant.
- Algorithms that allow different resistances to be distinguished:
    - Identification of the different color bands of the resistors;
    - Identification of resistors, allowing them to be distinguished from other electronic components present in the video.

Assessment

The following factors will be taken as evaluation criteria:

- Program quality:
  - development of the functionalities described in the work statement;
  - level of optimization of implemented functions;
  - correct functioning of the program;
  - added value [1].
- Quality of the code and respective comments, as well as the oral presentation:
  - correct and complete description of the program structure;
  - description of the techniques applied in the development of functionalities.
- Respect for work submission rules.
- Level of knowledge that the student has about the developed work.

The collective nature of the work does not affect the fact that the assessment is individual for each member of the group.

Each member of the group must demonstrate knowledge of all the code submitted.

Deadlines

Carrying out the work presupposes the delivery of files with the source code, in digital format. Do not submit the project folder, but only the source code files (`.c` / `.h` or `.cpp` / `.hpp`).

The work must be sent to the teacher via the submission link available on the Moodle page of this UC, by the defined date and time (also available on the Moodle page of the UC).

The delivery of practical work must comply with **the** following requirements:

- The source code files must be placed in a zip file with the name “**VC-TY-xxxx-xxxx-xxxx.zip**”, where:
  - **xxxx** must be filled in with the student number of each member of the group);
  - **TY** must be filled in with **T1** for Shift 1 groups and **T2** for Shift 2 groups.
- Only 1 (one) member of each group must submit the work.
- The name of the zip file should only contain the numbers of the students who actually carried out the work, and who, therefore, will be submitted to the assessment.

The delivery period ends on **June 1st, 2024**, at **11:00 pm**. **Works delivered after this date will not be considered**. Any change to the delivery and/or presentation date will be communicated to all students via Moodle.

[1] By added value we mean the way in which the work stands out (positively) from the rest.

Ethical conduct

The lack of transparency in assessments, whether in-person or otherwise, is naturally illegal and immoral. All sources used to support work must be clearly and clearly referenced. Any plagiarism, copying or inappropriate academic conduct will be penalized with the cancellation of the work. If there are noticeably similar works (where, for example, only the names of variables from another code have been changed) between groups, all similar works will be canceled.