Competition Task 4:
Image Processing and QR Code Identification

Objective of the task:
The task involves writing a Python application that will read QR code information from photos attached to the task. Participants should use an image processing library (e.g., OpenCV) and a dedicated QR code decoding library (e.g., pyzbar). The application should be able to load an image, process it appropriately, and display the QR code's contents on the screen. The program should utilize libraries such as OpenCV , PIL, numpy , tkinter / customtkinter , or other tools.
The files attached to the task are saved in jpg format.

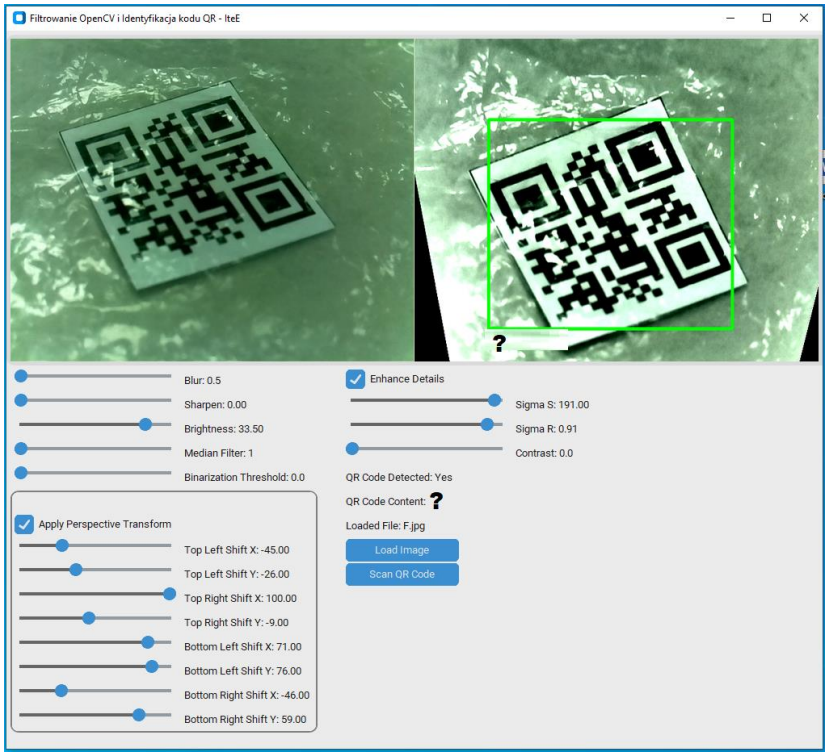

Fig. 1. Sample Python application

Task content:

1. **Libraries:** Design a Python application that can load and process images containing QR codes. Use libraries like OpenCV , PIL, numpy , tkinter / customtkinter , and pyzbar to process images and decode QR codes.
2. **Image Processing:** Process images to make the QR code easier to read (optional filters such as blur, sharpen, brightness/contrast correction, perspective transformation, binarization, enhance details or other).
3. **User Interface:** Design a simple yet functional user interface using tkinter / customtkinter that will allow you to load images, display the processed image and the QR code identification results.
4. **Documentation:** Participant should provide documentation of their code, explaining the key elements and operation of the program.

Evaluation criteria:

1. **Performance:** Effectiveness of reading information from various QR codes. Application performance when processing images. Resilience to errors after applying filtering. Implementation of additional filters and image processing settings.
2. **Code quality:** Cleanliness, structure, and comments in the source code.
3. **Interface functionality:** Intuitiveness and aesthetics of the graphical interface.
4. **Innovation:** Creative approach to the problem and possible additional features that distinguish the project.

Additional guidelines:

- The program should be written in Python and use the listed libraries.
- Code should be organized and easy to understand, with clearly defined functions and modules.
- Compatibility with the latest versions of the libraries used.
- The evaluation will take into account both the technical aspects of the solution and its practical usability.