

# Image Processing – Session 4

This session is devoted to the assignment you will need to handle by the date specified on Claco. You can choose any subject within the following list or make up your own. For each challenge, there are several levels; the higher the level reached, the better.

## Counting fingers and recognising gestures

With a single hand on screen, read the instructions it might be sending your way.

**Level 1** Detect and interpret the amount of fingers on screen.

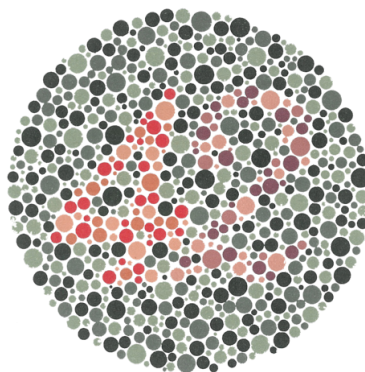
**Level 2** Detect and interpret the position of the fingers 🖐️👉👎

**Level 3** Detect and interpret the movement of the hand or fingers.

**Level Bonus** Play Rock-paper-scissors with the computer.

## Cheat the Ishihara Test

The Ishihara test purpose is to diagnose variations of colour blindness or deficiencies<sup>1</sup>. Your aim is to cheat the test using computer vision. Your program should be able to analyse various plates and detect its features.



**Level 1** Highlight the colours.

**Level 2** Change the colour scheme to black&white to highlight the number.

**Level 3** Make it working with a live video feed from the camera and superimpose the previous result on the actual image.

**Level 4** Recognise and translate the number from the test.

---

<sup>1</sup>Ishihara test on Wikipedia, complete test [here](#).

## Make up your own challenge

You have a great idea about a challenge involving computer vision and OpenCV ? Great, talk about it with a teacher to set the specifications and make sure you are within the expectations of the course.

## Evaluation

You should hand the following items in your submission:

- A report
- Your code
- A demonstration

Your report should be no longer than four pages long and have at least the following key points:

- Description of the context/situation chosen and expected results.
- Identification of the key challenges of the problem.
- Strategy to tackle the aforementioned challenges, how to implement a solution.
- Results and discussion of their quality.
- How to take it further.

Your code needs to be hosted on a versioning platform, such as GitHub or GitLab. The demonstration should be a recording of your screen showing the processing in action<sup>2</sup>. Upload it on a video hosting platform and add a link in your report. No *not* upload it on Claco or send it via email.

## The AI Question

Be it GitHub Copilot, Google Bard or OpenAI ChatGPT, you might be tempted to seek some support from a generative model, and that's all right! However you have to be transparent about it. If you use some code snippet from an online resource, you would need to cite the original work for adequate attribution and for verification. The same goes for language models: you need to acknowledge their use.

In practice, *if* you choose to use such tools, you need to:

1. Disclose it. Be as specific as you can about the version used.
2. Write a full write-up of your prompt engineering<sup>3</sup>. How did you make the model generate the parts you needed? How did you modify them? Did you use a specific context?

---

<sup>2</sup>Checkout OBS to record your screen.

<sup>3</sup>Example: GitHub repo, publication and authors interview

3. Write a paragraph about the criticism one could argue *against* the use of the system you used. Reference your arguments.

This extra material does *not* count toward the page count stated above.

Be warned that lack of transparency of whatever form or missing code attribution will be heavily penalised.