Technische Hochschule Ingolstadt
Fakultät Informatik

Computer Vision
Project Work

Dominik Rößle

18.05.2025

Project description

Project description

- Processing of video sequences (OpenCV Camera) and provision of the processed frames for a so-called virtual camera
  - OpenCV Camera provides the connection to your “real” plugged camera module
  - VirtualCamera acts like a container that renders an image sequence and provides an interface to most modern VoIP software, e. g. Zoom, Discord, etc.
- Groups of 3 students are formed
  - Each student should have the same time involvement in the presentation as well as in the execution of the project
- Presentation: 10 Minutes
  - Explain your approaches (especially the “special task”, see next slides)
  - Demo
- **Submission:** Your python code (don’t forget to use #code comments)

Tasks

Tasks

Must-have - Basics

- Basic operations:
  - Mean, Mode, Standard deviation, Max, Min
  - Linear transformation
  - Entropy
  - Histogram
    - For each channel (RGB) separately (three lines inside of one plot)
    - Equalization
  - Filter(s) of your choice (at least one)
    - Edge detection, blur, sharpen, sobel, gabor, etc.
    - No identity filters!

Tasks

Must-have – Something special

How do your results differ from the results of other groups?

- Examples of “special” (You can use libraries like cv2 for this task):
    - Object detection (e. g. face key points detection)
    - Detection of key points
        - Replace the shown image inside the key points and replace it with a different image
            - Face -> Dog/Cat/Emoji
            - Any object -> Any other replacement
    - Image segmentation tasks

You can also use neural networks to solve the “special” task. Due to the likely non-existent computing power, you are allowed to use pre-trained weights.

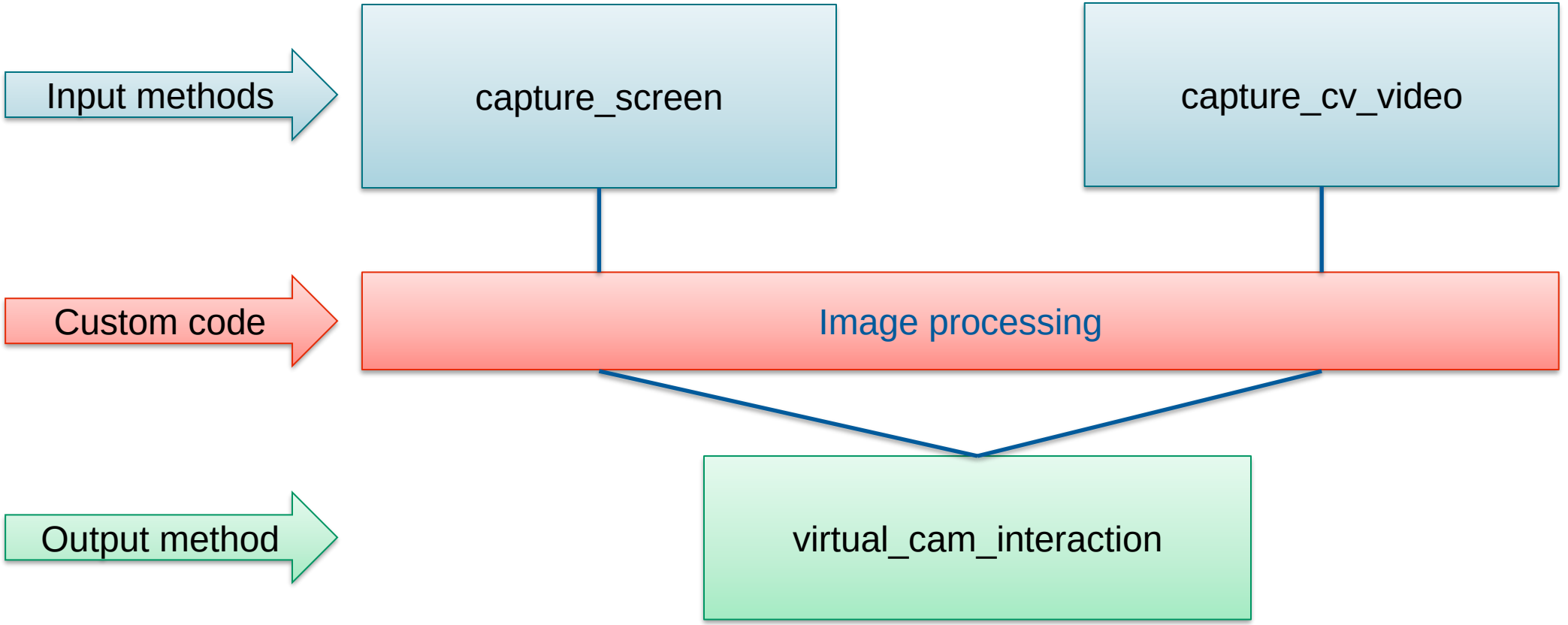
Demo

Software requirements

- We will use the python package **pyvirtualcam** for the project work
- Please make sure that you meet all requirements for your system. See **pyvirtualcam** for more information! Different requirements across operating systems!
- Download **OBS Studio** to test your implementations. This is a required step to run the scripts (Tested on Windows)

Code architecture

- Class VirtualCamera (capturing.py) - Responsible for data input and data output



Code architecture

Image processing

```
def main():
    # change according to your settings
    width = 1280
    height = 720
    fps = 30

    # Define your virtual camera
    vc = VirtualCamera(fps, width, height)

    vc.virtual_cam_interaction(
        custom_processing(
            vc.capture_cv_video(0, bgr_to_rgb=True)
        )
    )

if __name__ == "__main__":
    main()
```

Camera settings

Communication with virtual camera (Output)

Image processing (Your code)

Communication with “real” camera (Input)

Code architecture

Image processing – Your Code

```python
def custom_processing(img_source_generator):
    # STATES
    # VARIABLES
    # etc.

    for sequence in img_source_generator: #
        # DO SOMETHING WITH sequence
        # sequence is the image of the current frame

        sequence = sobel_filter(sequence)

        #...
        #...


        yield sequence
```

Your image processing code

Enables function loops inside virtual_cam_interaction, like