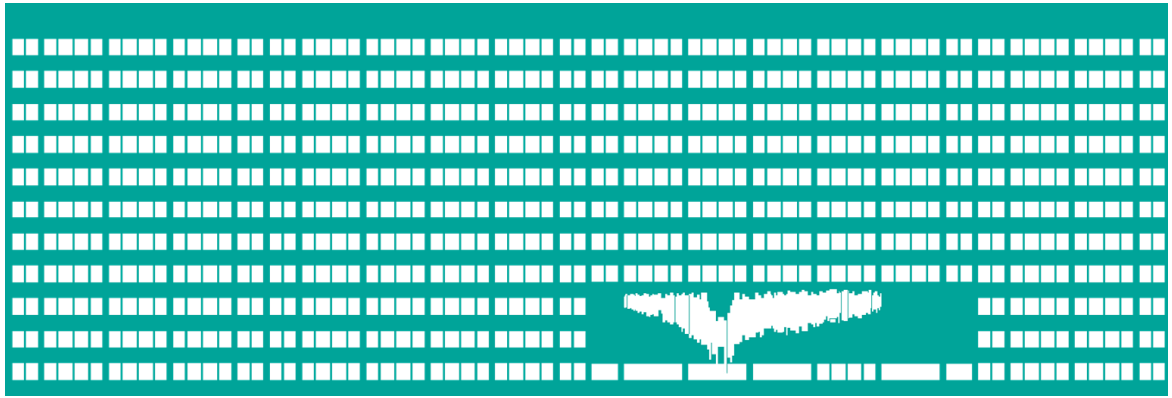

VŠB TECHNICKÁ
UNIVERZITA
OSTRAVA

VŠB TECHNICKÁ UNIVERZITA OSTRAVA

VSB TECHNICAL UNIVERSITY OF OSTRAVA



www.vsb.cz

Základy analýzy obrazu (ZAO)

Radovan Fusek

Intro

The most common tasks in image processing:

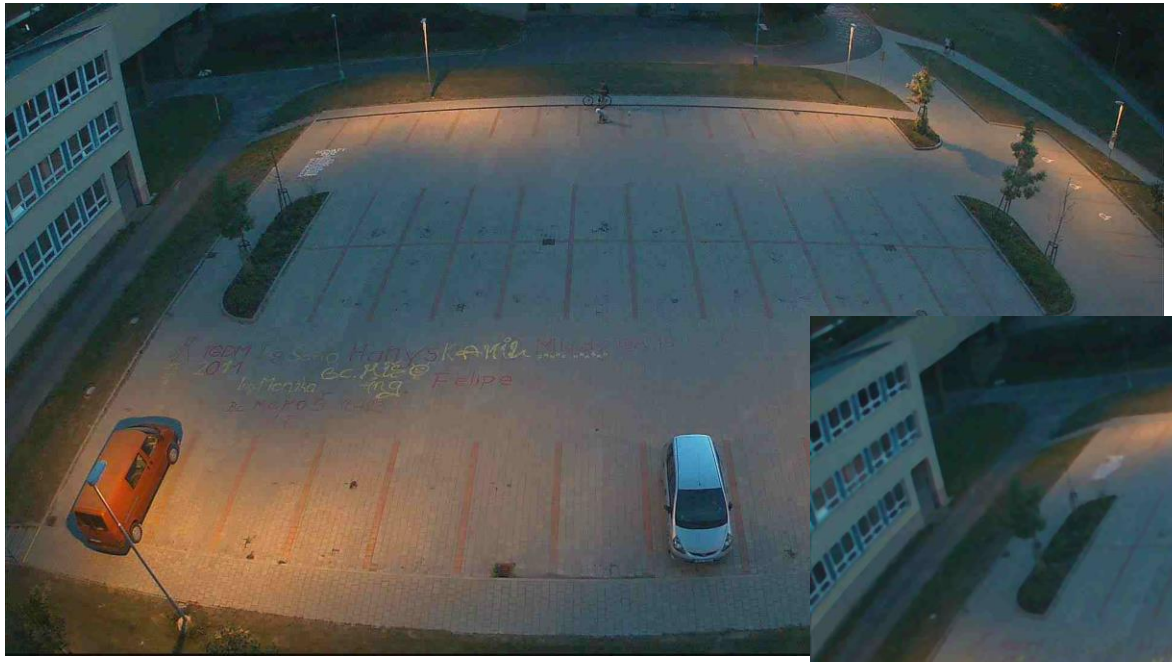
- Enhancement/Filtering

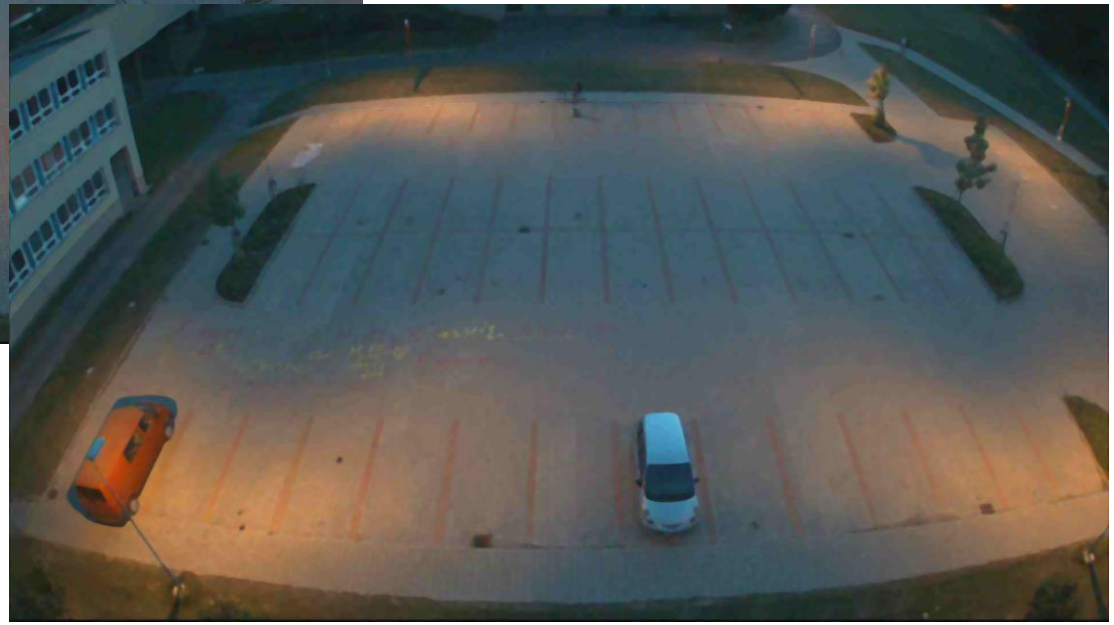
Filtering


EXPERIMENTAL
VEHICLE
FEIVSB02


EXPERIMENTAL
VEHICLE
FEIVSB02

Filtering



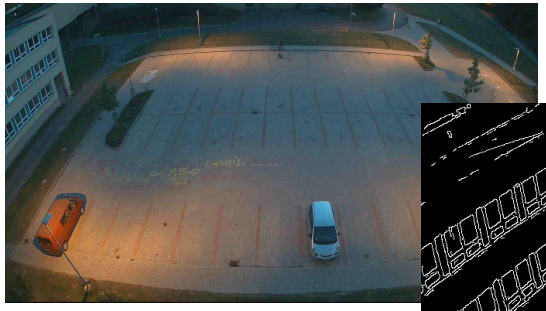


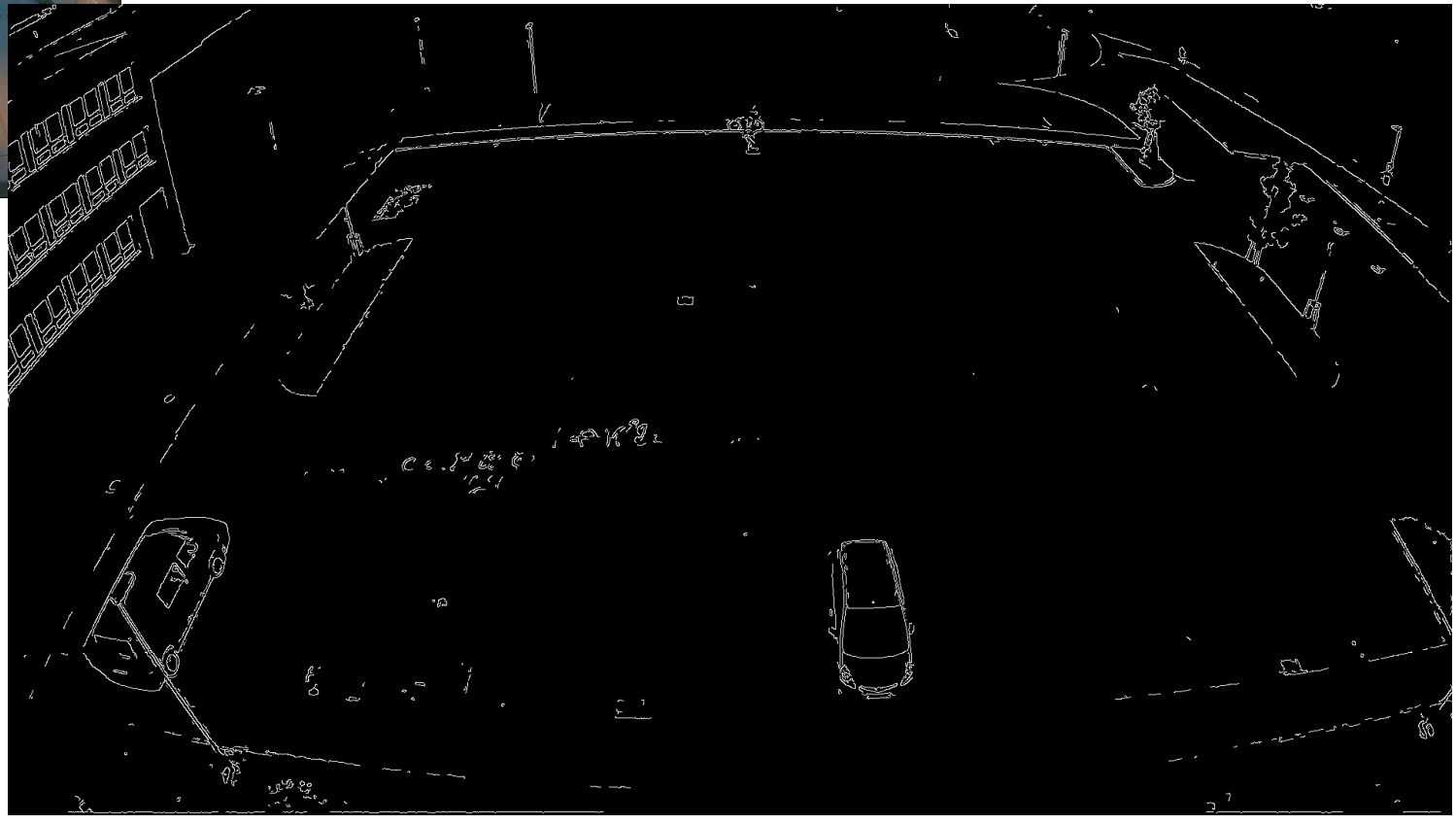
Intro

The most common tasks in image processing:

- Enhancement/Filtering
- Edge Detection

Edge Detection





Edge Detection

Video Example

Intro

The most common tasks in image processing:

- Enhancement/Filtering
- Edge Detection
- Object Detection

Object Detection

What is goal of object detection methods?

Object Detection

What is goal of object detection methods?

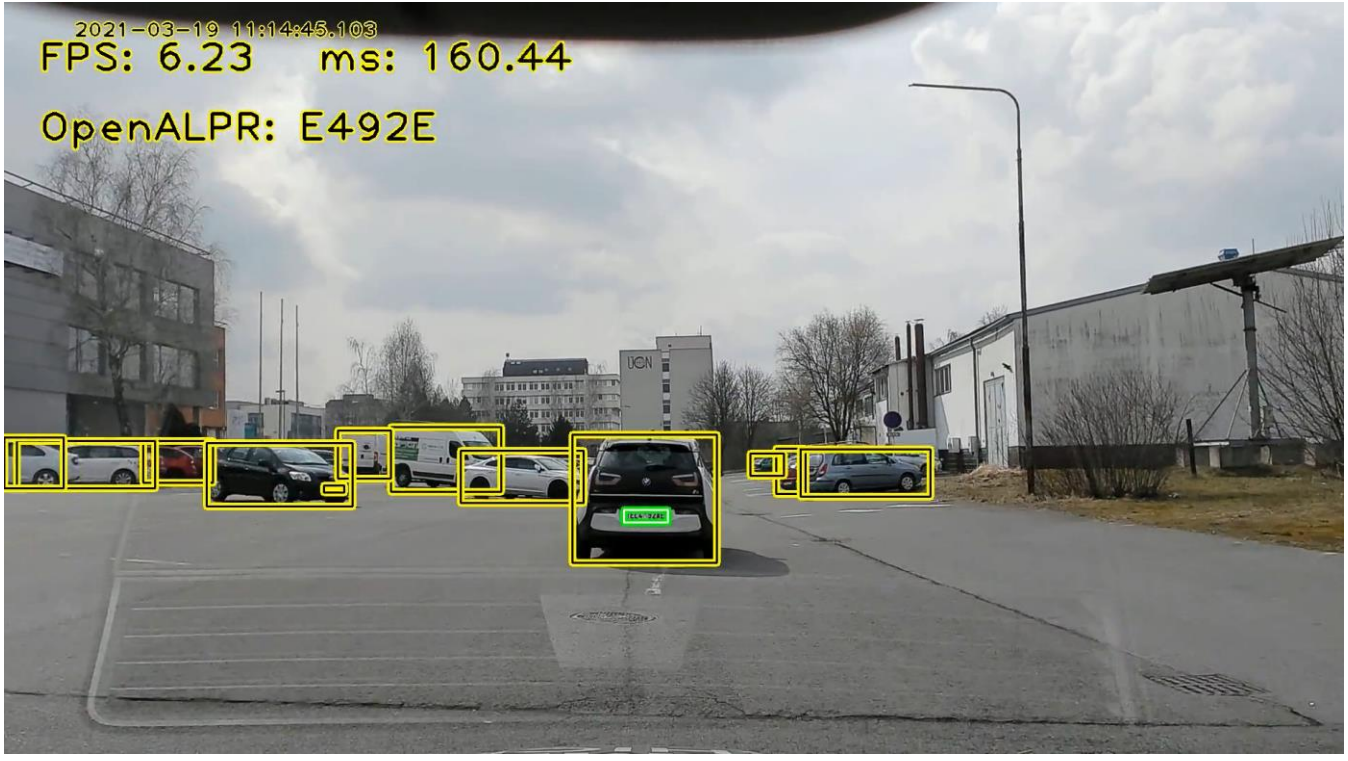
- It is clear that the images contain many objects of interest. The goal of the object detection systems is to find the location of these objects in the images (e.g. cars, faces, pedestrians).

- For example, the vehicle detection systems are crucial for traffic analysis or intelligent scheduling, the people detection systems can be useful for automotive safety, and the face detection systems are a key part of face recognition systems.

Object Detection

What is goal of object detection methods?

- position of the objects (cordinates) + scale of the objects



Object Detection

Video Example

Intro

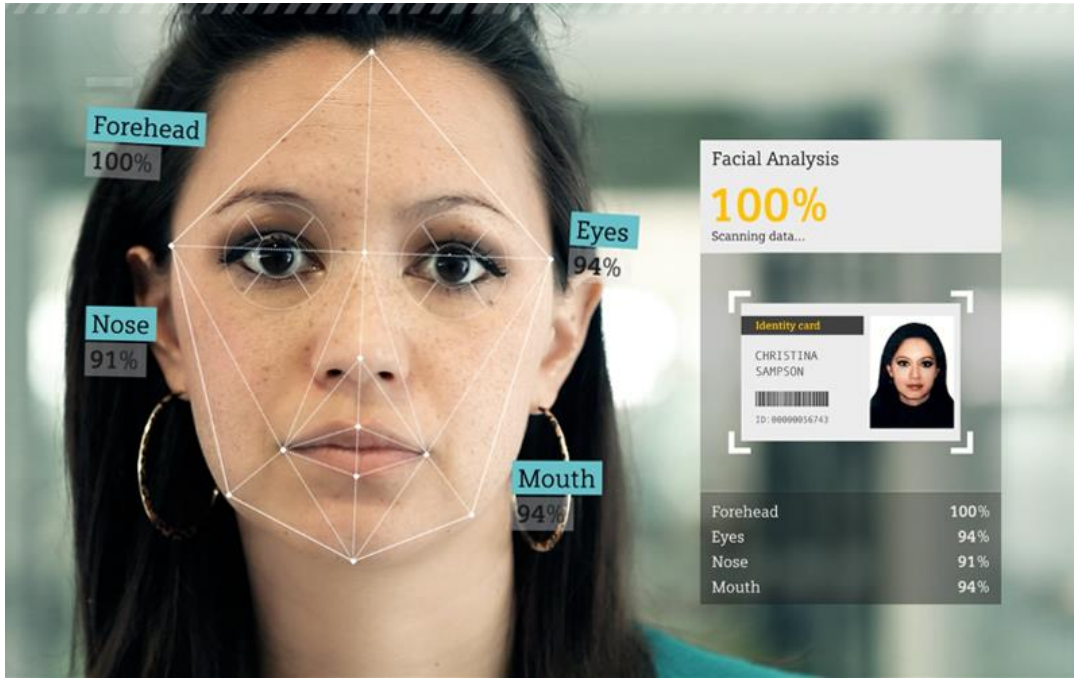
The most common tasks in image processing:

- Enhancement/Filtering
- Edge Detection
- Object Detection
- Object Recognition

Object Recognition

What is goal of object recogniton methods?

- identifying objects (categories)
- e.g. face recognition



https://recfaces.com/articles/how-accurate-is-facial-recognition

Intro

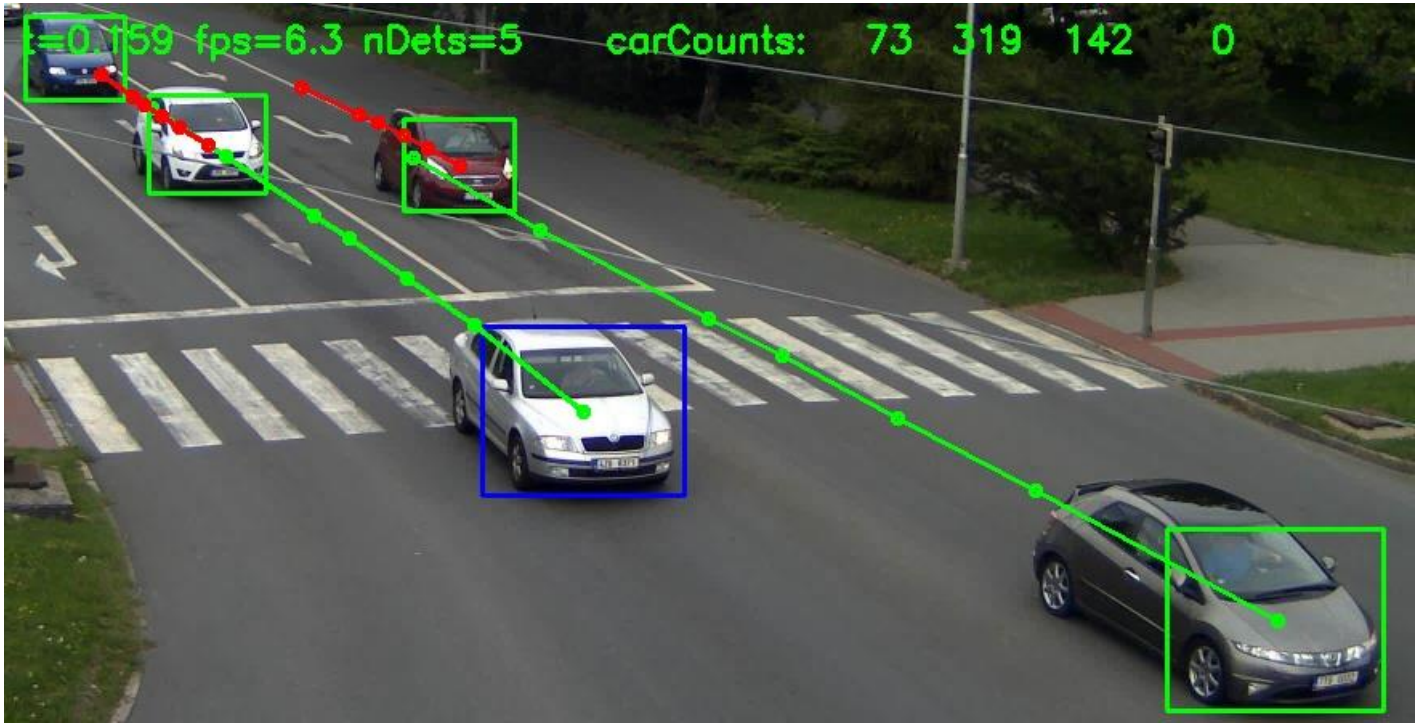
The most common tasks in image processing:

- Enhancement/Filtering
- Edge Detection
- Object Detection
- Object Recognition
- Object Tracking

Object Tracking

Object Tracking

- e.g. estimation of trajectory/unique ID to each tracked object



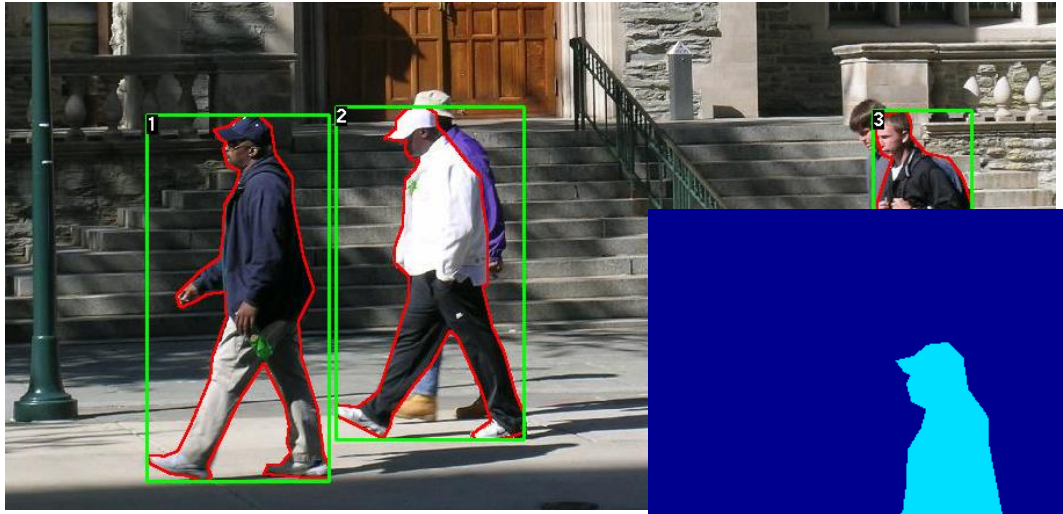
Intro

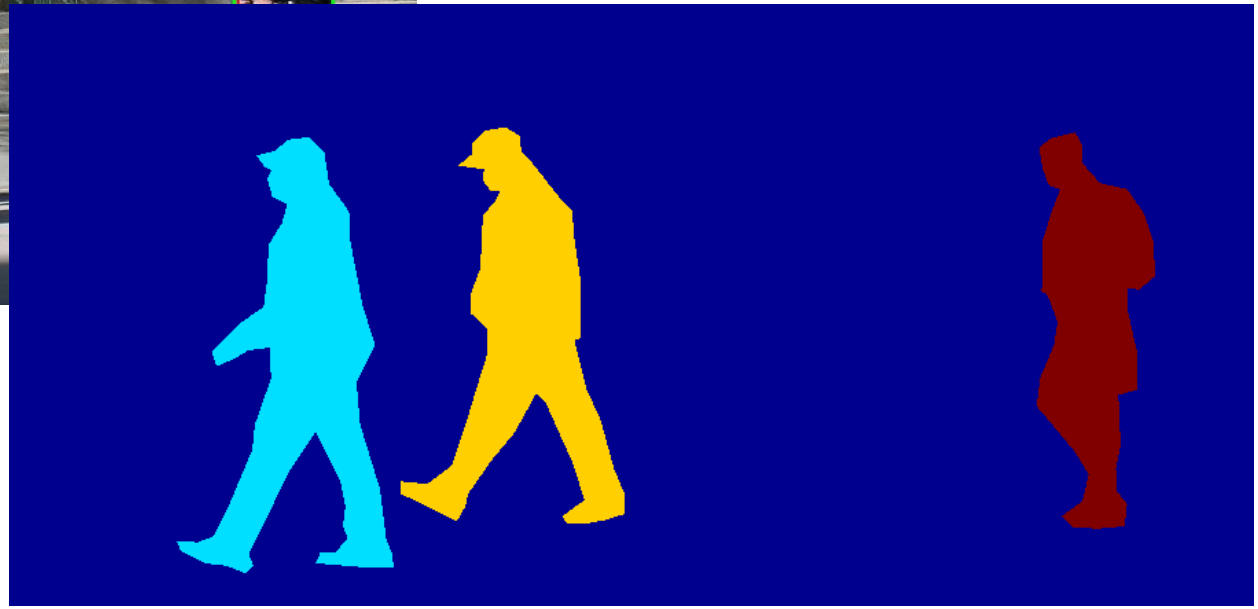
The most common tasks in image processing:

- Enhancement/Filtering
- Edge Detection
- Object Detection
- Object Recognition
- Object Tracking
- Segmentation

Object Segmenation

Image is partitioned into multiple regions





Object Segmenation

Image is partitioned into multiple regions

Object Segmenation

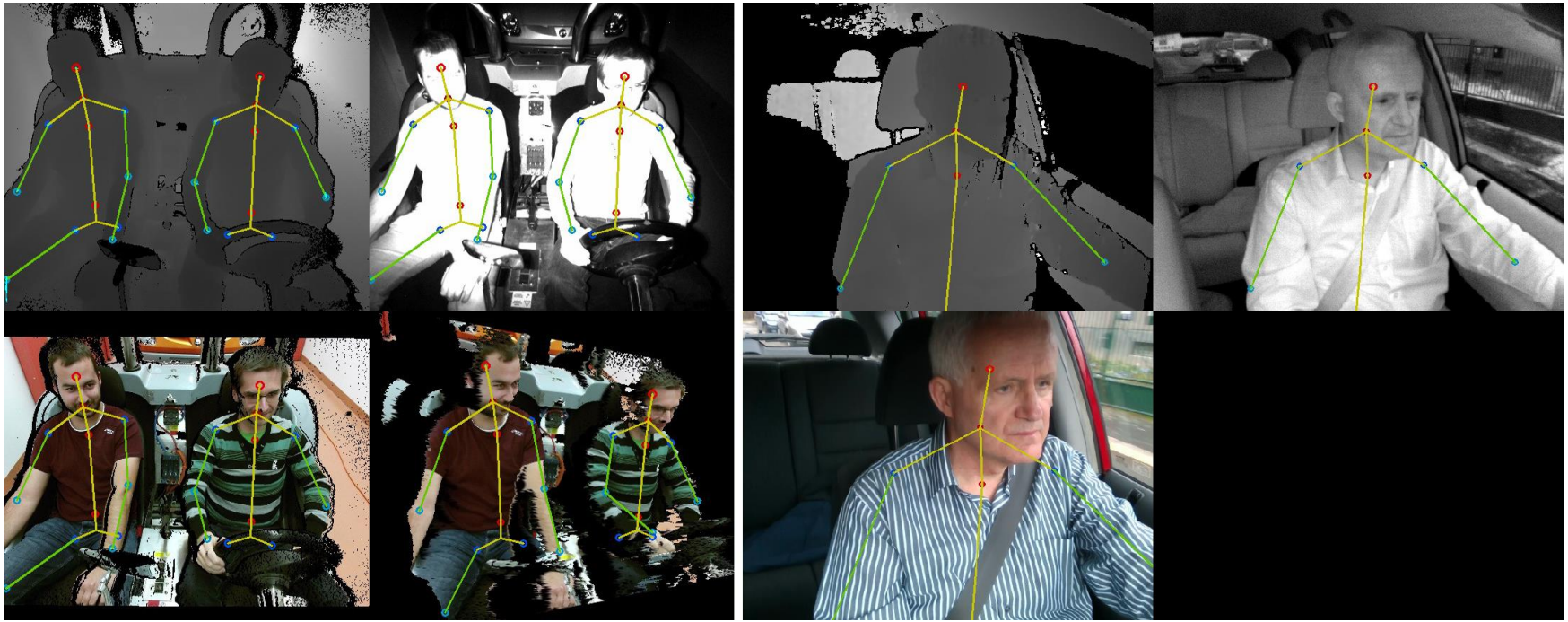
Video Example

Intro

The most common tasks in image processing:

- Enhancement/Filtering
- Edge Detection
- Object Detection
- Object Recognition
- Object Tracking
- Segmentation
- Human Pose Detection
- ... many others

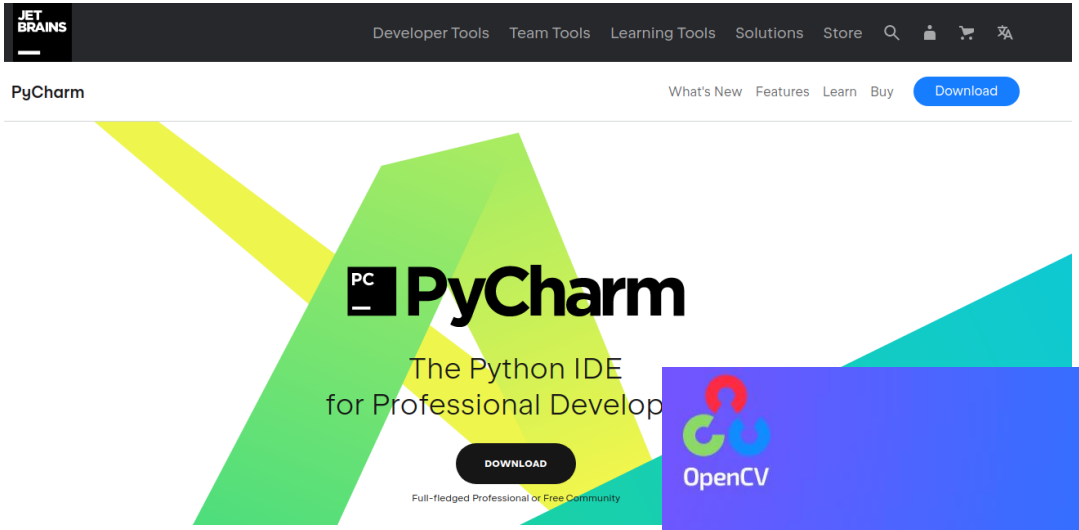
Pose Detection

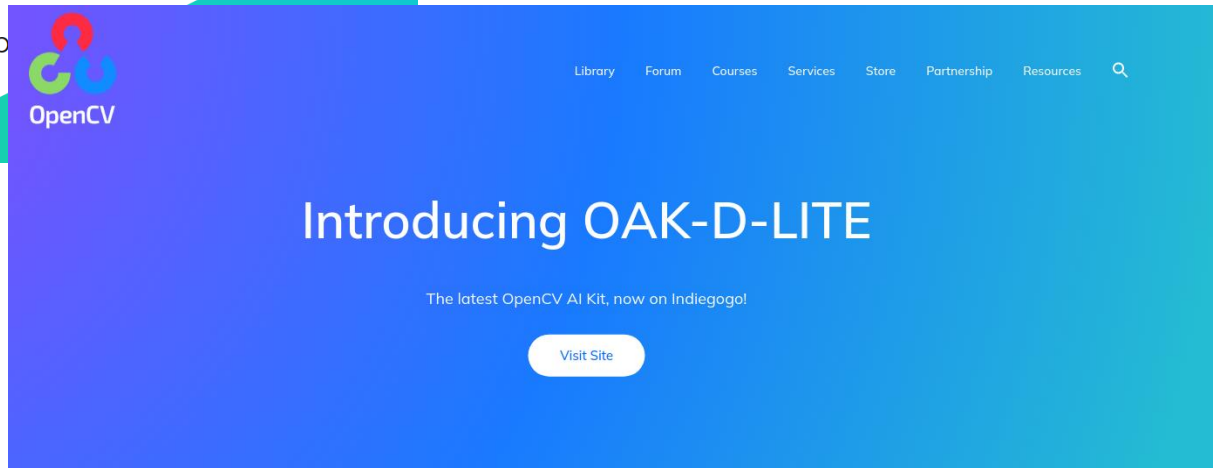


Pose Detection

Video Example

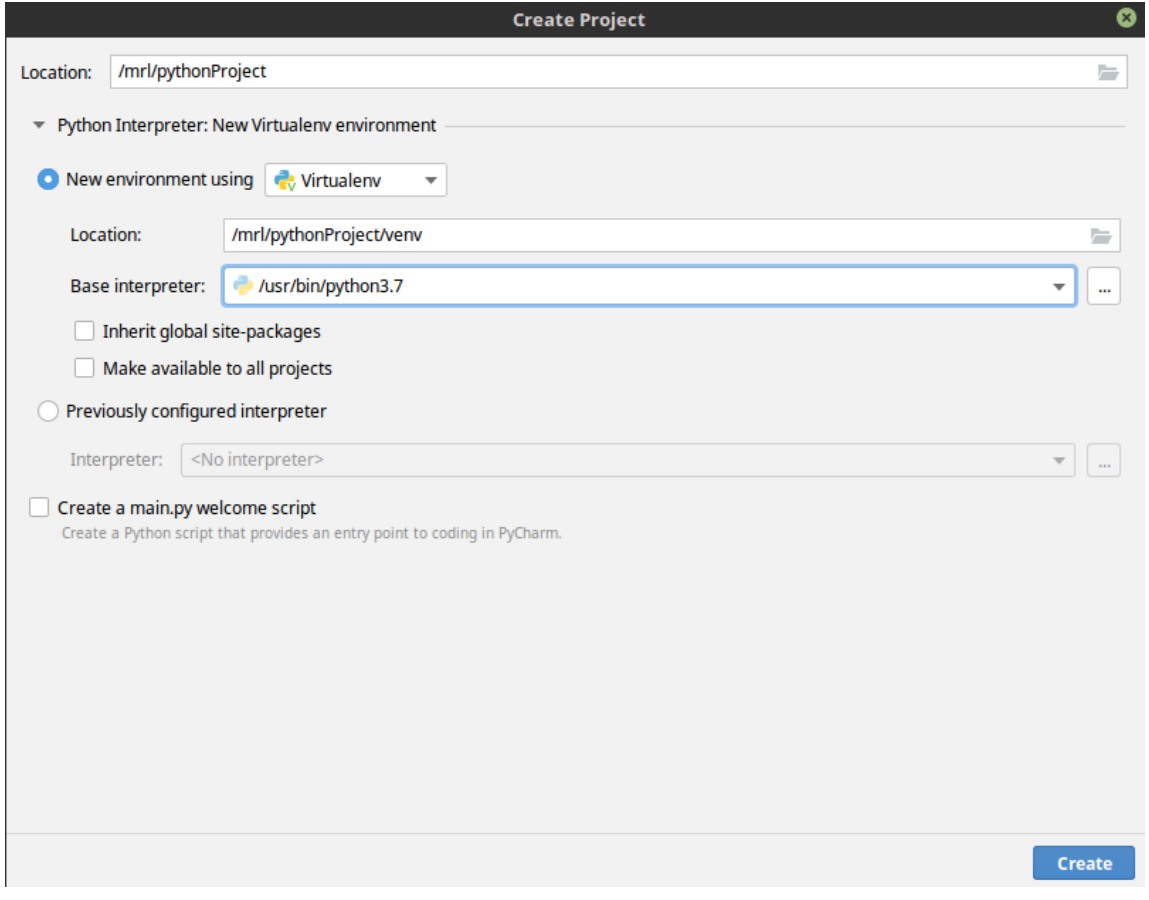
Opencv + Python





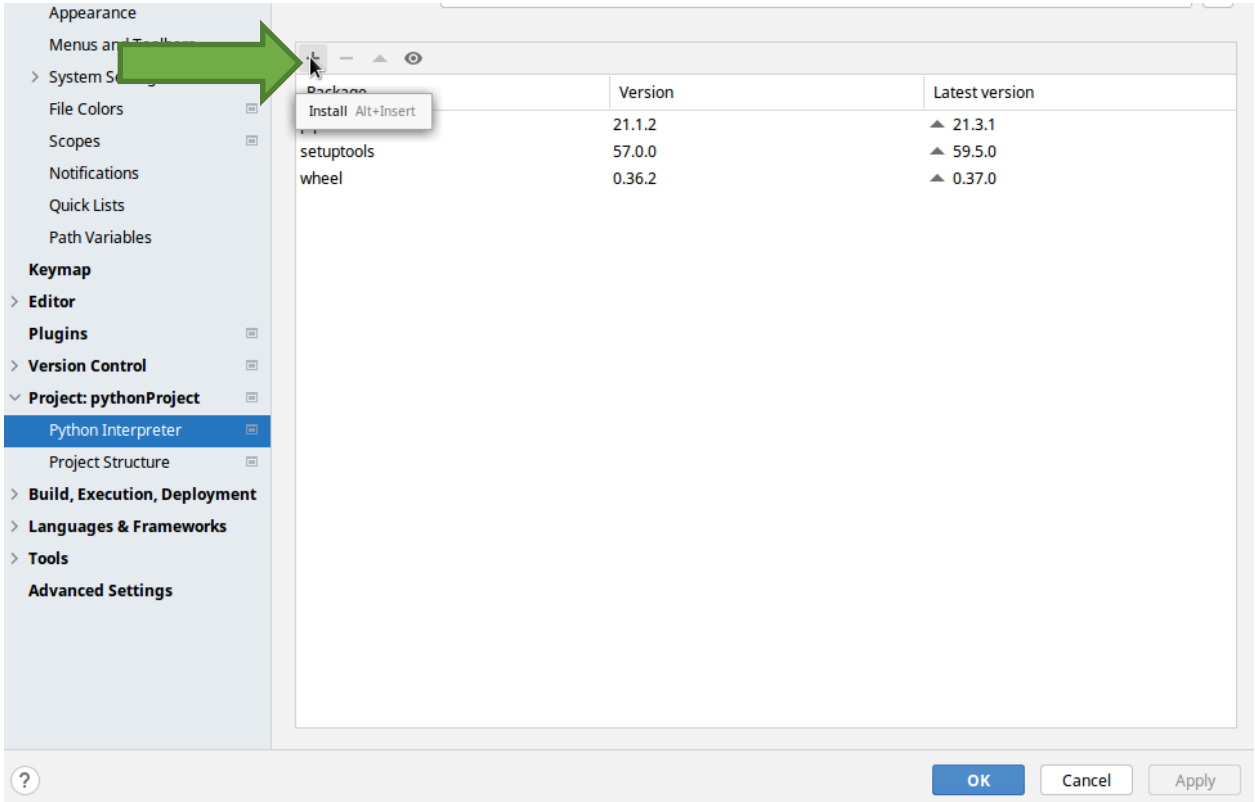
python-3.7.9-amd64
(OK with Dlib)

Opencv + Python

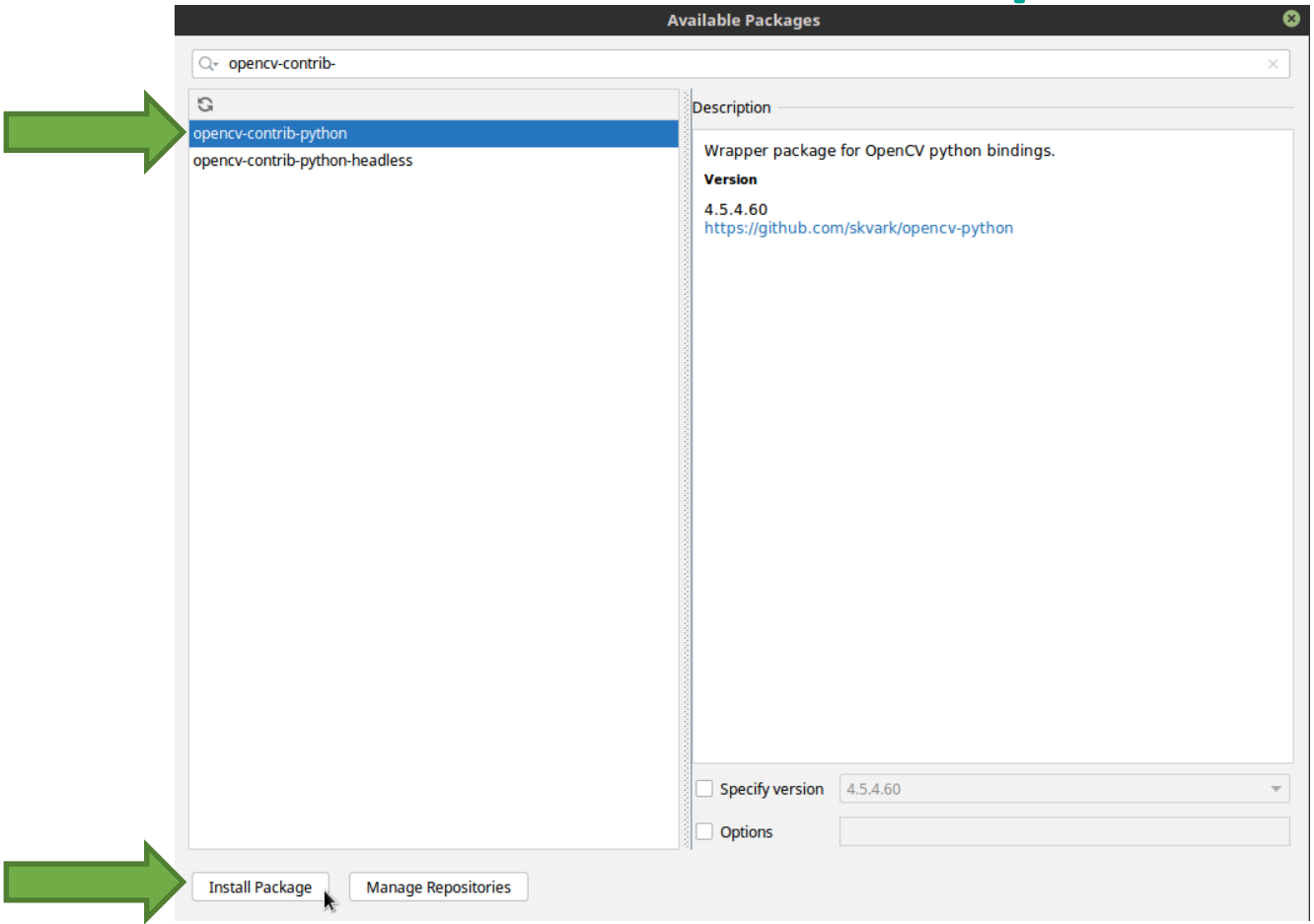


Opencv + Python

Ctrl+Alt+S



Opencv + Python

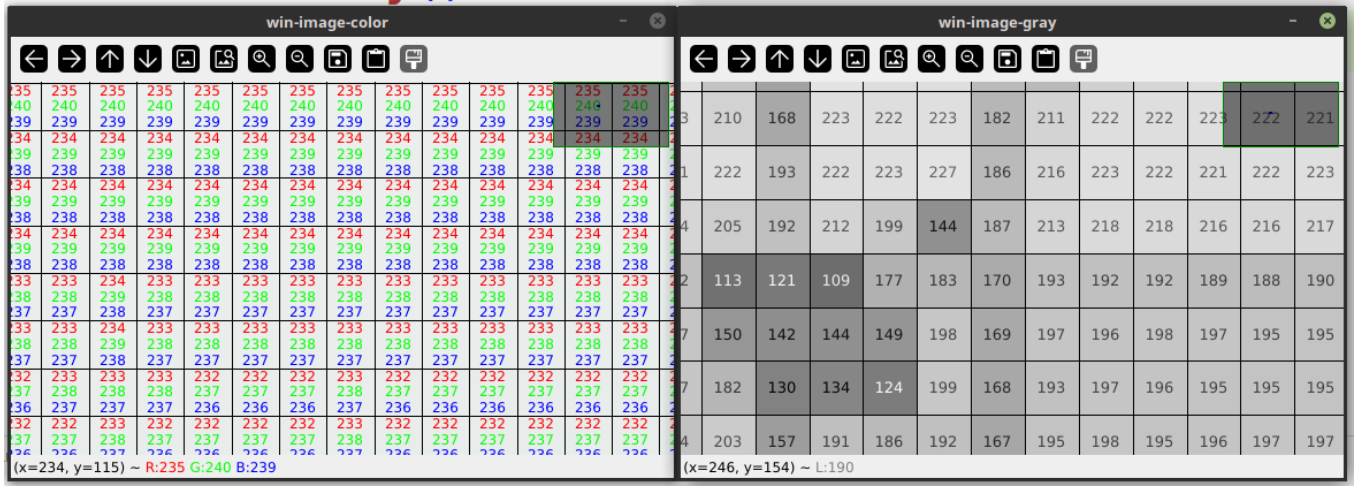


Opencv + Python

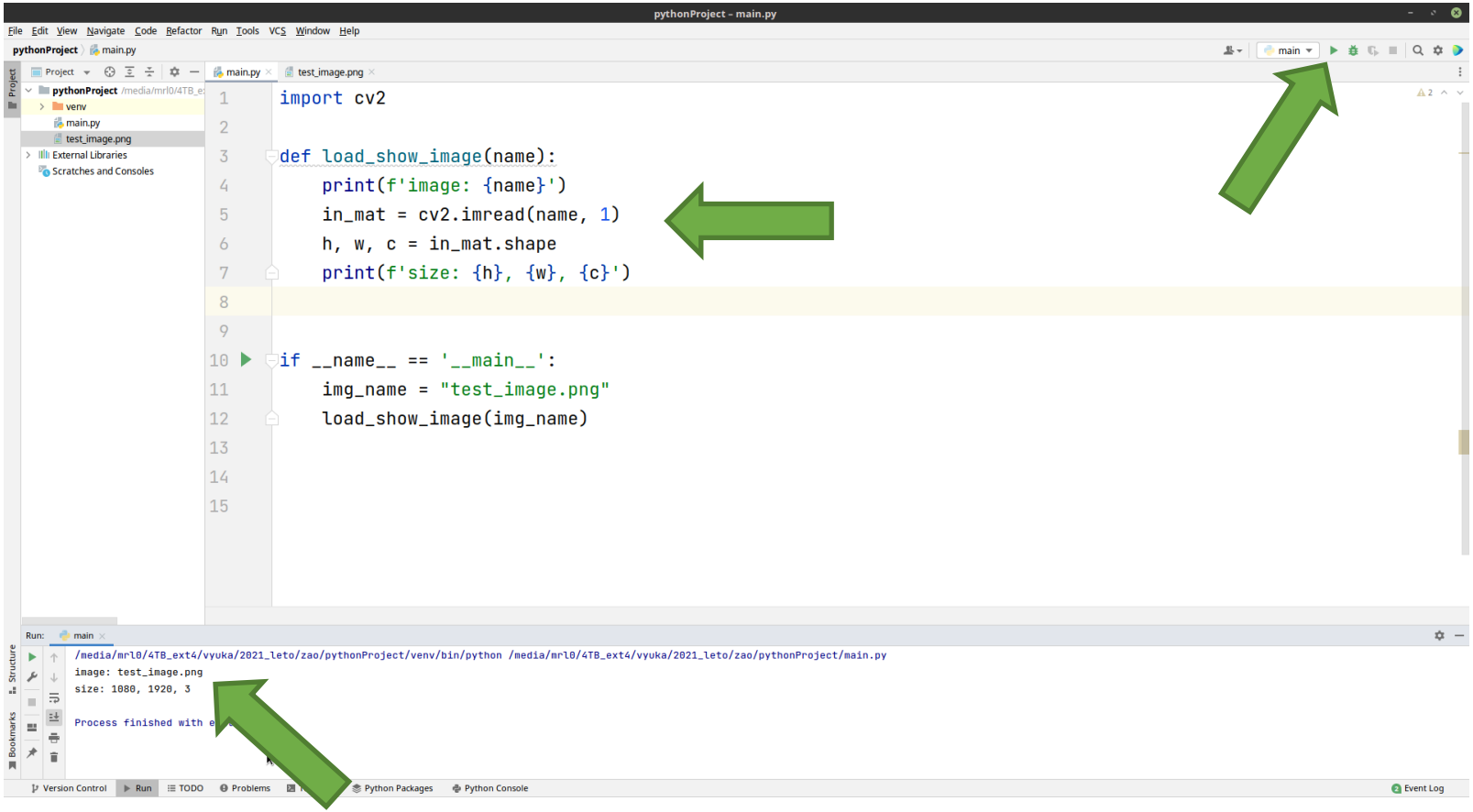
```python
import cv2

image_gray = cv2.imread("img.png", 0)
image_color = cv2.imread("img.png", 1)
cv2.imshow("win-image-gray", image_gray)
cv2.imshow("win-image-color", image_color)
cv2.waitKey()
```

Opencv + Python

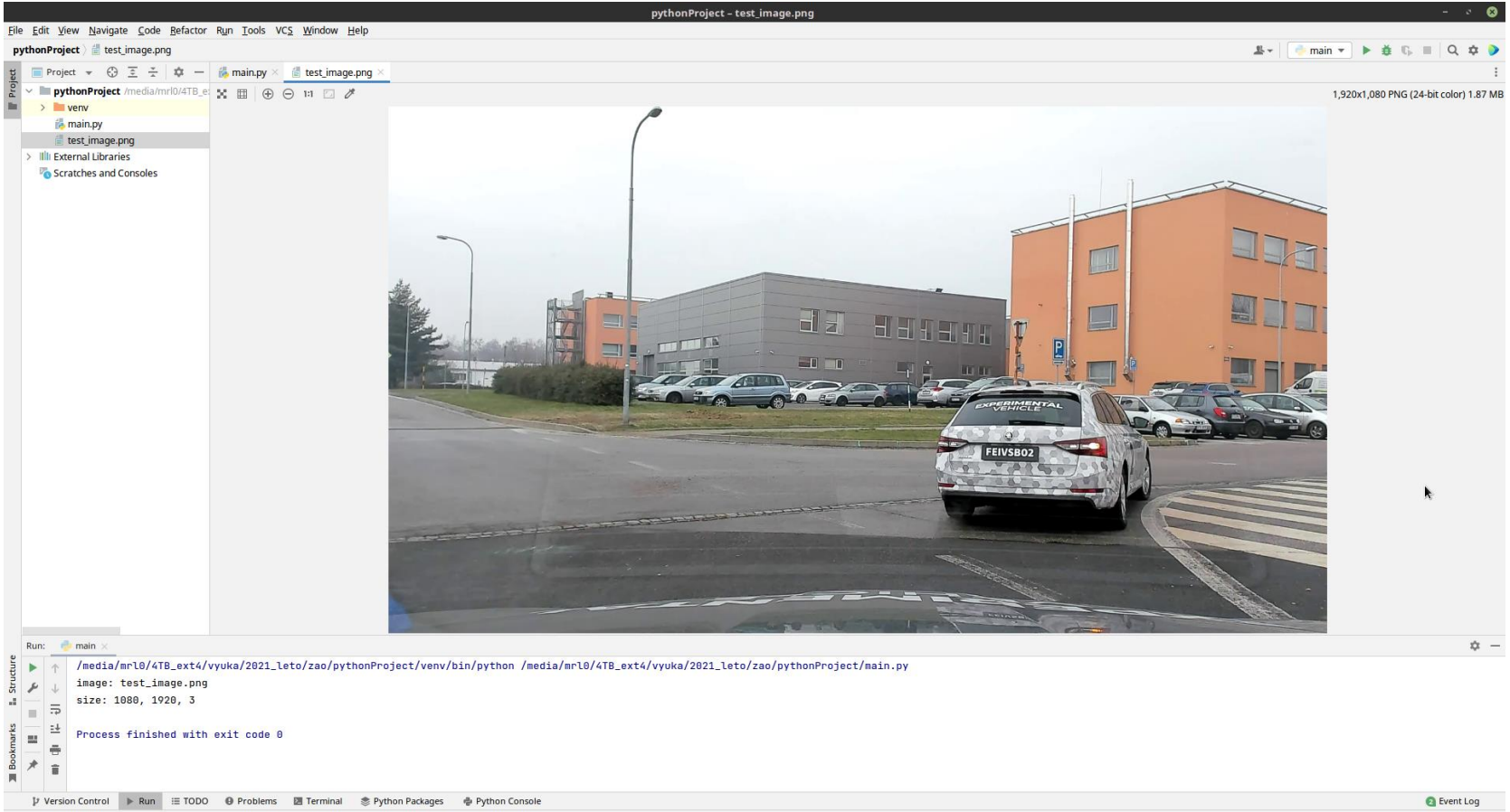
```python
import cv2

image_gray = cv2.imread("img.png", 0)
image_color = cv2.imread("img.png", 1)
cv2.imshow("win-image-gray", image_gray)
cv2.imshow("win-image-color", image_color)
cv2.waitKey()
```



Opencv + Python



Opencv + Python



Opencv + Python

pythonProject – main.py

File Edit View Navigate Code Refactor Run Tools VCS Window Help

pythonProject › main.py

Project: pythonProject /media/mrl0/4TB_e…
- venv
- main.py
- test_image.png
- External Libraries
- Scratches and Consoles

main.py | __init__.py | test_image.png

```
1  import cv2
2
3
4  def load_show_image(name):
5      print(f'image: {name}')
6      in_mat = cv2.imread(name, 1)
7      h, w, c = in_mat.shape
8      print(f'size: {h}, {w}, {c}')
9
10     win_name = "image"
11     cv2.namedWindow(win_name, 0)
12     cv2.resizeWindow(win_name, int(w/3), int(h/3))
13     cv2.imshow(win_name, in_mat)
14     cv2.waitKey(0)
15
16
17 if __name__ == '__main__':
18     img_name = "test_image.png"
19     load_show_image(img_name)
20
21
```

Run: main

```
/media/mrl0/4TB_ext4/vyuka/2021_leto/zao/pythonProject/venv/bin/python /media/mrl0/4TB_ext4/vyuka/2021_leto/zao/pythonProject/main.py
image: test_image.png
size: 1080, 1920, 3
```

Version Control | Run | TODO | Problems | Terminal | Python Packages | Python Console | Event Log

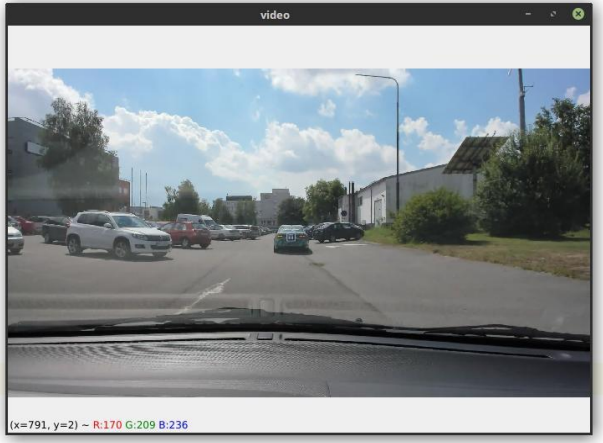
Opencv + Python

pythonProject – main.py

File Edit View Navigate Code Refactor Run Tools VCS Window Help

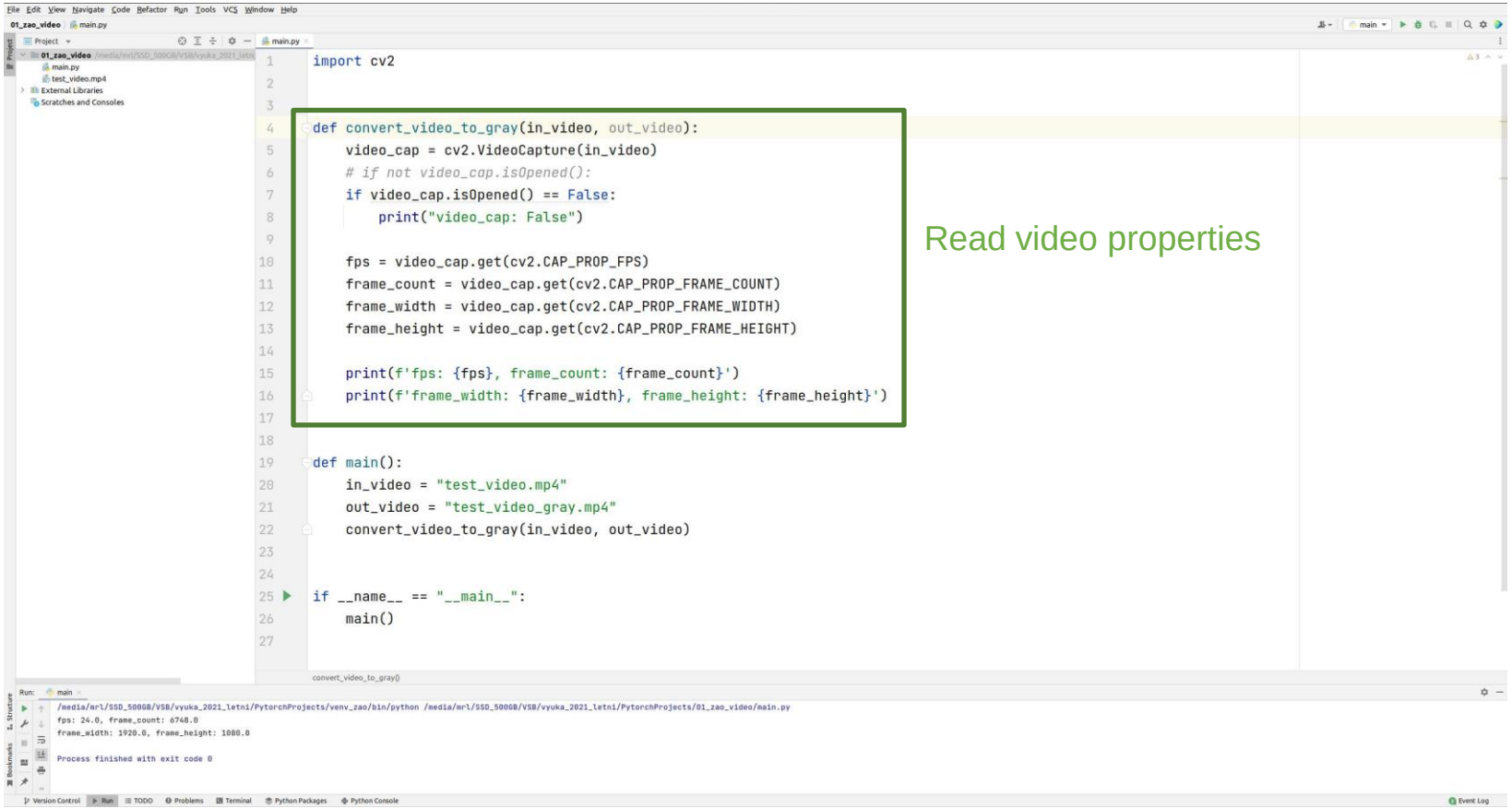
pythonProject / main.py

```
def load_show_video(name):
    video_cap = cv2.VideoCapture(name)
    win_video = "video"
    cv2.namedWindow(win_video, 0)
    while True:
        ret, frame = video_cap.read()
        if ret is True:
            cv2.imshow(win_video, frame)
            if cv2.waitKey(1) == ord('q'):
                break


if __name__ == '__main__':
    img_name = "test_image.png"
    vid_name = "test_video.mp4"
    #load_show_image(img_name)
    load_show_video(vid_name)
```




```
QApplication: invalid style override 'gtk' passed, ignoring it.
    Available styles: Windows, Fusion
```

Opencv + Python



Read video properties

Opencv + Python

```python
def convert_video_to_gray(in_video, out_video):
    video_cap = cv2.VideoCapture(in_video)
    # if not video_cap.isOpened():
    if video_cap.isOpened() == False:
        print("video_cap: False")

    fps = video_cap.get(cv2.CAP_PROP_FPS)
    frame_count = video_cap.get(cv2.CAP_PROP_FRAME_COUNT)
    frame_width = video_cap.get(cv2.CAP_PROP_FRAME_WIDTH)
    frame_height = video_cap.get(cv2.CAP_PROP_FRAME_HEIGHT)

    print(f'fps: {fps}, frame_count: {frame_count}')
    print(f'frame_width: {frame_width}, frame_height: {frame_height}')

    frame_size = (int(frame_width), int(frame_height))
    video_writer = cv2.VideoWriter(out_video,
                                   cv2.VideoWriter_fourcc('M', 'J', 'P', 'G'),
                                   fps,
                                   frame_size,
                                   False)
    frame_num = 0
    while (video_cap.isOpened()):
        ret, frame = video_cap.read()
        if ret:
            frame_gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
            video_writer.write(frame_gray)
            frame_num = frame_num + 1
            print("frame_num: ", frame_num)
```

Save video

VideoWriter() [2/3]

cv::VideoWriter::VideoWriter (const **String** & filename,
int fourcc,
double fps,
Size frameSize,
bool isColor = `true`
)

Python:

cv.VideoWriter( ) -> <VideoWriter object>
cv.VideoWriter(filename, fourcc, fps, frameSize[, isColor]) -> <VideoWriter object>
cv.VideoWriter(filename, apiPreference, fourcc, fps, frameSize[, isColor]) -> <VideoWriter object>

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Parameters

filename	Name of the output video file.
fourcc	4-character code of codec used to compress the frames. For example, **VideoWriter::fourcc**('P','I','M','1') is a MPEG-1 codec, **VideoWriter::fourcc**('M','J','P','G') is a motion-jpeg codec etc. List of codes can be obtained at Video Codecs by FOURCC page. FFMPEG backend with MP4 container natively uses other values as fourcc code: see ObjectType, so you may receive a warning message from OpenCV about fourcc code conversion.
fps	Framerate of the created video stream.
frameSize	Size of the video frames.
isColor	If it is not zero, the encoder will expect and encode color frames, otherwise it will work with grayscale frames (the flag is currently supported on Windows only).

Tips:

- With some backends `fourcc=-1` pops up the codec selection dialog from the system.
- To save image sequence use a proper filename (eg. `img_%02d.jpg`) and `fourcc=0` OR `fps=0`. Use uncompressed image format (eg. `img_%02d.BMP`) to save raw frames.
- Most codecs are lossy. If you want lossless video file you need to use a lossless codecs (eg. FFMPEG FFV1, Huffman HFYU, Lagarith LAGS, etc...)
- If FFMPEG is enabled, using `codec=0; fps=0;` you can create an uncompressed (raw) video file.

Opencv + Python

```python
def convert_video_to_gray(in_video, out_video):
    video_cap = cv2.VideoCapture(in_video)
    # if not video_cap.isOpened():
    if video_cap.isOpened() == False:
        print("video_cap: False")

    fps = video_cap.get(cv2.CAP_PROP_FPS)
    frame_count = video_cap.get(cv2.CAP_PROP_FRAME_COUNT)
    frame_width = video_cap.get(cv2.CAP_PROP_FRAME_WIDTH)
    frame_height = video_cap.get(cv2.CAP_PROP_FRAME_HEIGHT)

    print(f'fps: {fps}, frame_count: {frame_count}')
    print(f'frame_width: {frame_width}, frame_height: {frame_height}')

    frame_size = (int(frame_width), int(frame_height))
    video_writer = cv2.VideoWriter(out_video,
                                   cv2.VideoWriter_fourcc('M', 'J', 'P', 'G'),
                                   fps,
                                   frame_size,
                                   False)
    frame_num = 0
    while (video_cap.isOpened()):
        ret, frame = video_cap.read()
        if ret:
            frame_gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
            video_writer.write(frame_gray)
            frame_num = frame_num + 1
            print("frame_num: ", frame_num)
```

Save video

VideoWriter() [2/3]

cv::VideoWriter::VideoWriter (const **String** & filename,
int fourcc,
double fps,
Size frameSize,
bool isColor = `true`
)

Python:

cv.VideoWriter( ) -> <VideoWriter object>
cv.VideoWriter(filename, fourcc, fps, frameSize[, isColor]) -> <VideoWriter object>
cv.VideoWriter(filename, apiPreference, fourcc, fps, frameSize[, isColor]) -> <VideoWriter object>

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Parameters

filename	Name of the output video file.
fourcc	4-character code of codec used to compress the frames. For example, **VideoWriter::fourcc**('P','I','M','1') is a MPEG-1 codec, **VideoWriter::fourcc**('M','J','P','G') is a motion-jpeg codec etc. List of codes can be obtained at Video Codecs by FOURCC page. FFMPEG backend with MP4 container natively uses other values as fourcc code: see ObjectType, so you may receive a warning message from OpenCV about fourcc code conversion.
fps	Framerate of the created video stream.
frameSize	Size of the video frames.
isColor	If it is not zero, the encoder will expect and encode color frames, otherwise it will work with grayscale frames (the flag is currently supported on Windows only).

Tips:

- With some backends `fourcc=-1` pops up the codec selection dialog from the system.
- To save image sequence use a proper filename (eg. `img_%02d.jpg`) and `fourcc=0` OR `fps=0`. Use uncompressed image format (eg. `img_%02d.BMP`) to save raw frames.
- Most codecs are lossy. If you want lossless video file you need to use a lossless codecs (eg. FFMPEG FFV1, Huffman HFYU, Lagarith LAGS, etc...)
- If FFMPEG is enabled, using `codec=0; fps=0;` you can create an uncompressed (raw) video file.