

# Telepresence robot, nano-computers and advanced cameras as educational tools

Julien Marot

Aix Marseille Université, Institut Fresnel  
Marseille, France  
julien.marot@fresnel.fr

Michel Bensoam

Aix Marseille Université, Lab. Info Systemes  
Marseille, France  
michel.bensoam@univ-amu.fr

**Abstract**—This paper describes a project course and a classroom course, performed in collaboration with the firm Axyn Robotics, where students aim at improving the autonomy of a telepresence robot. In this research, firstly, the students had to get familiar with the robot and control it remotely with WiFi and bluetooth communication protocols. Secondly, the students had to explore the abilities of new advanced cameras constructed by Intel. They had to show to the industrial partner that the chosen cameras were adequate to enable the robot to perform autonomous navigation. A classroom lesson is proposed where three parts of the robot are studied through smaller setups. We strongly believe that the measured positive academic impact of such a project and course is due to the university-industry collaboration, and the use of information and communication technologies as educational tools.

**Index Terms**—education, robotics, cameras, university-industry collaboration

## I. INTRODUCTION

Nano-computers and artificial intelligence are currently must-know topics for Masters students in electrical engineering and automation. We wanted to exemplify these topics with a central item which is a telepresence robot constructed by an industrial partner. A telepresence robot, essentially, should enable mobile video-conference and can be displaced remotely. A challenging topic in this field is the transfer of such an advanced technology into the teaching domain.

### **Relation to prior work in the field:**

In [2], the authors aim at improving computer science education through algorithm visualization. But data processing and communication technologies are not reduced to algorithms. It is important to relate software issues to a hardware support, such as a moving robot. The technology acceptance model (TAM) [1] states that people are eager for novel technologies if they understand their usefulness. Hence the need for an application when teaching informatics and artificial intelligence (AI) [5]. The TurtleBot for instance, has been used to teach basics about ROS (Robot Operating System).

The TurtleBot can localize itself and drive around a room. Also, it can be controlled remotely from a laptop, or an Android-based smart phone [3]. However, we could notice that it may be difficult for the students to get familiar with ROS. Hence we preferred to work with other operating systems such as Raspbian, which is closer to Linux.

In [10], an AI flavored Interaction is proposed with robots,

over Wi-Fi. The considered robots are similar to remote-controlled cars. This work has much merit but it lacks in terms of concrete cooperation with a firm. We wanted to involve the students in the development of a robot which is currently sold in the market. We had the opportunity to work with the firm Axyn Robotics, which is close to our university. Axyn Robotics has put effort in building a friendly interface for its telepresence robot, and aims at purchasing inexpensive software and hardware items to propose a robot which can be widely used for reduced costs. At the university side, we also encourage the use of cheap hardware setups for project courses and practicals [13]. Raspberry Pis for instance are rather inexpensive, and permit though to create and run somehow advanced digital image processing algorithms [16], [18].

### **Main contributions:**

Firstly, we propose a project course which involves the industrial partner, and relies on nano-computers and advanced cameras sold by Intel®. This permits to introduce the students with a rich system involving a hardware part, software through OpenCV programmes, and a basic application of industrial internet of things [4]. The Intel cameras [6] are compatible with a multitude of computers, including Personal Computers and the Odroid XU4Q [14]. The latter is a very potent nanocomputer already used by Axyn for other purposes, and whose processing power would allow the implementation of complex algorithms.

Secondly, we propose a classroom lesson and a series of practicals to teach nano-computers through the example of the robot.

### **Outline:**

In section II we present our setup and student tasks in the frame of a project course. Its academic impact is evaluated in the form of a student questionnaire.

In section III we present a way to use the robot and a series of setups in a classroom course and practicals. A 'puzzle' approach is adopted where students work on low-cost setups representing the 'head' and the 'feet' of the robot.

## II. PROJECT COURSE

This project course results from a series of issues that the industrial partner and our university wish to solve jointly. We are particularly interested in improving the autonomy of the

telepresence robot while moving in an open-space such as a flat. For instance, as illustrated in Fig. 1, the robot may have to move from the charging station in the living room, to the kitchen, avoiding a sofa, and finding the doors.



Fig. 1. Example of trajectory: from the charging station to the kitchen

For this, we wish to use machine vision for autonomous navigation. The robot has to be able to receive instructions remotely, to perceive its environment in order to avoid obstacles, and to locate itself in his environment. With a view to complete these objectives, we decided to provide the students with the hardware and software described in the next subsection.

#### A. Hardware

We afford a customized robot (called Ubbo) and two advanced cameras, both provided by the industrial partner.

1) *Robot*: The different versions of Ubbo telepresence robots incorporate several types of nano-computers like Arduinos, Raspberries and Odroid XU04. We can see the latest version of the Ubbo Maker [17] in Fig. 2. The students in the project course were provided with an open and custom version of the robot, displayed in Fig. 3.



Fig. 2. Ubbo Maker robot

The 'head' of the customized robot is composed of a Raspberry Pi, a Pi camera, and a touchscreen display.

Figure 5 shows one of the student groups at the industrial partner, during a presentation of the Ubbo robot.



Fig. 3. Customized Ubbo Maker robot

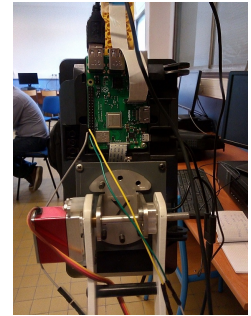
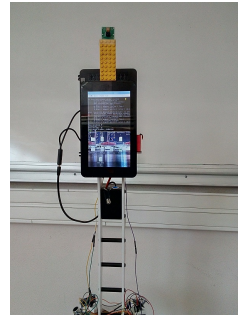


Fig. 4. Head of the customized robot



Fig. 5. Students at Axyn

2) *Cameras*: Part of our project course is based on two cameras proposed since January 2018 by Intel [6]: the depth camera D435i and the tracking camera T265. RealSense technology is made of vision processors, depth and tracking cameras, supported by an open source, cross-platform software development kit called 'librealsense'. The extensions and instructions for installations are available at [7], [8]. With these cameras we aim to produce and run useful codes for our final goal, that is, improving the self localization of the robot. The depth camera can perform object recognition and distance detection. The tracking camera can map a trajectory between two points, in our example, the charging station and the kitchen. Therefore, the features of both the depth camera D435i and the tracking camera T265 are useful. The idea then is to allow the robot in the future to memorize some useful paths and use the tracking camera for self localization along these paths. The depth camera should be used to avoid obstacles and improve the robustness of the system with

respect to modifications in the environment. The cameras are shown in Fig. 6.



D435i Depth Camera



Tracking Camera T265

Fig. 6. Real Sense Intel cameras

### B. Operating system, softwares and programmes

As the principles of Internet of Things are now must-know for our students, we wished that the students in this project control remotely the robot. For this, the students used their smartphone, and two free applications: one from Axyn to move the robot, and another to control a Raspberry Pi which is in the head of the robot. Adapting the Intel cameras to improve the autonomy of the robot is the most innovative part of the project course. In this paper we describe the experiments which were performed by the students to get familiar with the cameras. As for now, experiments are performed on a laptop. The experiments are detailed in subsubsection II-B2.

1) *Remote control of the robot:* A specific smartphone application has been developed by Axyn, called 'Ubbo Maker', available on Google play for Android. It permits to control remotely the Arduino nano-computer which controls the wheels of the robot, with a bluetooth connection. A Raspberry Pi can be remotely controlled through WiFi signals, with free smartphone applications such as 'Bluetooth Terminal Emulator', 'Hackers keyboard', 'RaspiCheck', etc. We chose RaspiCheck, because it is free, and because we did not need to emulate a keyboard. Moreover, the students noticed that it was adequate to launch python programmes which had been stored in advance in the Raspberry Pi.

While providing the students with two applications, the first one using bluetooth, and the second one using WiFi, we also wanted to introduce them with various communication protocols. An 'Apache' server has been created by the students to transfer documents, such as snapshots and outputs of face detection algorithms, from the head of the robot to a Laptop.

2) *Intel cameras: experiments and results obtained by the students:* To make experiments with the Intel cameras, the students used personal computers with the latest version of Windows 10 installed, as well as Visual Studio 2015 and all extensions downloadable on the Intel cameras website [6]. Sample codes are available at [9]. The students figured out that

the cameras accept a multitude of languages, and that most codes are currently written in C++ with a heavy emphasis in object oriented programming. The students also noticed that several programmes required OpenCV. A significant help must be provided by the teacher to the project students so that they create new codes. The four most interesting codes are explored in the following.

*Depth Perception with Intel Viewer Software:* The left and right imagers of the camera capture the scene and send the data to the image processing unit, which calculates the depth for each pixel by comparing the two images. A depth frame is generated with the calculated values. The infrared projector can also be used, to increase the contrast and allow the camera to operate in low light conditions. It is important to say that the camera calculates the depth with respect to a parallel plane of the imagers and not the absolute range. Figure 7 exemplifies the type of output image that can be provided by the D435i Depth camera. It includes an RGB sensor which provides the image above.



Fig. 7. Example of acquisition obtained with the depth camera

It is interesting to notice that the closest spots to the camera are shown in black, meaning that no distance data were acquired for this spots. Also, the person which is the most distant to the camera almost disappears, since it is represented in a tone of red which is similar to the background.

*The object recognition algorithm:* As described in the documentation for the code, it is possible to use Intel RealSense cameras with existing Deep Neural Network algorithms. The demo is derived from the MobileNet Single-Shot Detector example provided with OpenCV. The original algorithm for mobile object detection is available at [15]. It is modified to work with Intel RealSense cameras and take advantage of depth data (in a very basic way). It is able to recognize some

everyday objects, like a person or a chair, and compute their distance to the camera; which is useful for self localization and obstacle avoidance in a flat. It uses pre-existing OpenCV data to perform object recognition. An example of its recognition abilities is provided in Fig. 8. A limitation of this system is also emphasized: although persons in the picture are far from the objective, and others are close, they appear equidistant. This shows that the depth perception of this camera is based on stereovision. Although very complete, the code is relatively slow, and its use should be restricted to the cases where the recognition of objects is really necessary. We also could think about remote computations on distant servers.

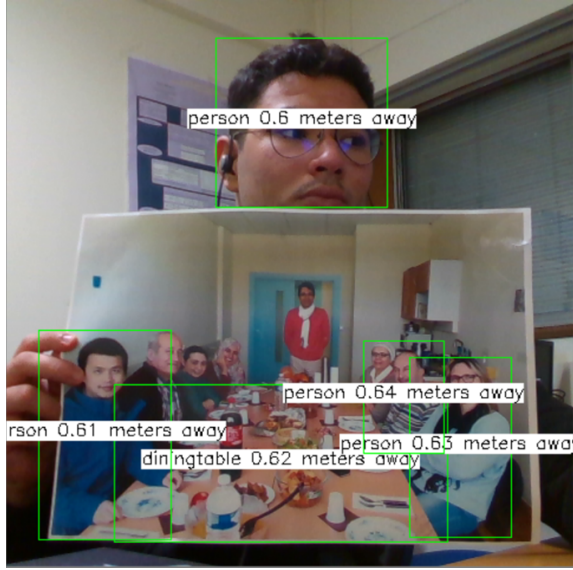


Fig. 8. Example of object recognition using the Depth Camera

*The distance detection algorithm:* This algorithm allows a precise measurement of the distance of an object located exactly at the center pixel of the image captured by the depth camera. This fact constitutes a limitation of the code for obstacle detection in the case of an object which is not located in the center of the camera's field of view. This was later corrected by the students with the implementation of a 25 point analysis, with these points scattered throughout the acquired image. The output of this algorithm is a decimal number with the distance in metres.

*The tracking algorithm:* This algorithm is implemented in the tracking camera and permits to record the trajectory of the robot.

The students have implemented a storing, along time in '.txt' files, of the spatial coordinates of the tracking camera. A Further work will consist in using these files in a semi-supervised learning phase where the robot will 'learn to know' his environment, and a non-supervised phase where the robot will have to move autonomously from one point to another in his environment, such as a flat, and for instance from the living room to the kitchen and conversely.

### C. Academic impact of the project course

At the end of the project course, all of the students completed an evaluation to determine their level of satisfaction in different areas (methodology, materials, and overall learning experience). They belong to five different MsCs: electrical engineering (first and second year), Europhotonics, signal and image processing (first and second year). In this section, we summarize the results. Their satisfaction was rated with four possible answers:  $A$ =No (1), rather no (2), rather yes (3), yes (4). We chose the same questions as in [10], which are relevant for our course as well. In addition, question 1 tells what Master of Science (MsC) the students belong to. The number of students  $N_s$  who participated is  $N_s = 14$ .

- 1) What is the MsC you belong to ?
- 2) Rate the contribution of the proposed techniques to better understand hardware issues (networking, motor driving, mechanical parts assembling, etc.).
- 3) Rate the contribution of the proposed techniques to better understand software issues (network programming, operating systems, AI, interface design, etc.).
- 4) Rate the contribution of the proposed techniques to better understand scientific field issues provided by your university curriculum.
- 5) Rate the contribution of the proposed techniques to become better in collaboration issues.
- 6) Rate the contribution of the proposed techniques to become better in problem solving issues.
- 7) Rate the contribution of the proposed techniques to become better in presenting a topic in public.
- 8) Assess the degree of completion of the assigned tasks by your team, at the end of the project.
- 9) Assess the degree of the cooperation among the members of your team.
- 10) Rate to what extent you are satisfied of your peers, assisting you to better understand the issues needed to be accomplished.
- 11) Rate to what extent you would like to keep the proposed activities in the university lessons curriculum.
- 12) Rate to what extent the technologies and practices being used would promote your career.
- 13) Assess your previous experiences using similar devices and practices in your student life.

Let  $Nb_i^A$  denote the number of students who answered  $A$  to question  $i$ . The percentage of students who answered the value  $A$  to question  $i$  ( $i = 1, \dots, 13$ ) is defined by:

$$P_i = 100 * \frac{Nb_i^A}{N_s} \quad (1)$$

The percentage values  $P_i$ ,  $i = 1, \dots, 13$  are presented in Table I. 'NR' means not relevant. This study was performed with  $N_s = 14$  students.

The detail of the evaluation, performed with the specific online evaluation tool 'Ametice' of our university, is available at [11]. From Table I, we notice that the answers are mostly 'rather good' (3) or 'good' (4). There exists an exception for



Question index	1	2	3	4
1	NR	NR	NR	NR
2	21	14	43	21
3	0	0	29	71
4	0	0	50	50
5	0	0	29	71
6	0	0	29	71
7	7	7	14	71
8	0	0	29	71
9	7	7	57	29
10	7	21	14	57
11	0	14	43	43
12	7	0	21	71
13	0	7	21	71

TABLE I

EVALUATION OF THE PROJECT COURSE: DISTRIBUTION OF THE ANSWERS TO QUESTIONS 1 TO 13, IN %.

Question 2: we believe that the students of the 'signal and image' MSc considered that they did not study the automotive part of the robot. Which is true because they were rather asked to work on the Raspberry Pi which constitutes the head of the robot.

For a teacher in the field, proposing such a project course allows him to become familiar with the robot through student projects. We now propose a layout for a classroom lesson and practicals, whose principles could be transferred to another robot or system.

### III. CLASSROOM LESSON AND PRACTICALS

A course syllabus is described here. We present the learning objectives, the education process, and a satisfactory survey. Our overall objective is that the students understand how the robot works. It is also to let them think about an adequate manner to improve the autonomy of the robot.

The overall didactic approach is a 'puzzle' process: The robot is decomposed into its main parts and the students work on these parts through practicals.

What the students must do follows the order of the subsections III-A and III-B.

#### A. Definition of a strategy

Firstly we organize a course with a demo involving the robot. In this way the students start wondering how it works, ask general questions about the structure of the robot, the motors, hardware, etc.

We explain the students what is our final goal: propose a two step methodology for the robot to move autonomously. We remind that the first step is a semi-supervised learning phase, and that the second phase is non-supervised. We ask the students to propose a strategy, and write it down without having to put it into operation.

In their report they have to:

- define the task the robot should accomplish in the first and second steps;
- design a strategy involving all sensors and nano-computers to allow the robot to move autonomously, from the charging station in the living-room to the kitchen and return;

- choose and place adequately the cameras and nano-computers for this purpose.

Due to their cost, only one or two robots can be used for the demo, and not all students in a practical group can use it as a whole. We propose a series of practical works.

#### B. Autonomous work on practicals

Secondly, we let the students work on setups. Each setup represents one part of the robot. The robot is 'decomposed' into its main parts: the 'head' and the 'feet'. Each pair of student affords a setup. Two setups are proposed, for a duration of 8 hours for each setup. Each setup can be duplicated easily because it is relatively cheap. In addition to these setups, the students perform a programming practical, involving remote computations.

1) *The head of the robot*: The objective is to perform the detection of face elements with a Raspberry Pi coupled with a camera (see Fig. 9).

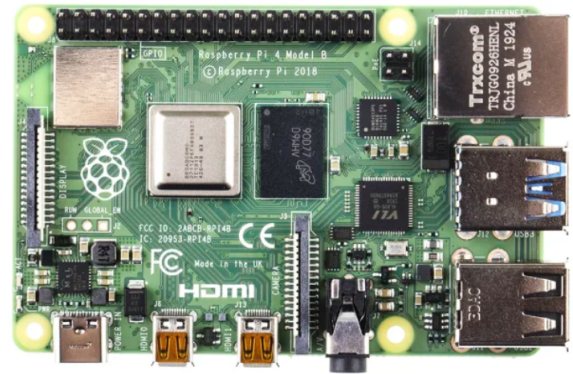


Fig. 9. Illustration of setup 1: Raspberry Pi model 4

2) *The feet of the robot*: The objective is to test an Arduino on a small electrical motor (see Fig. 10);

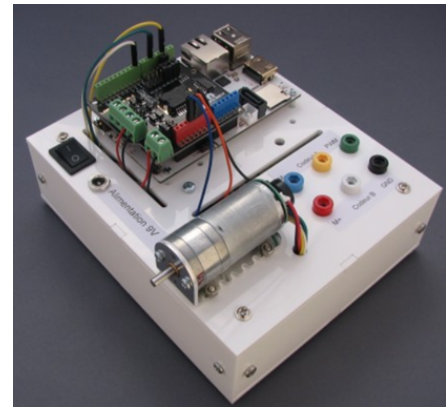


Fig. 10. Illustration of setup 2: Control of a small motor with an Arduino nano-computer

3) *Remote computations*: As mentioned previously, the robot could be assisted by remote servers to recognize objects. To illustrate this, we propose a third practical to the students, involving python programmes on Google Scholar. Fig. 11

illustrates this with a screenshot of a code for face detection. We measured the impact of these lesson and practicals.

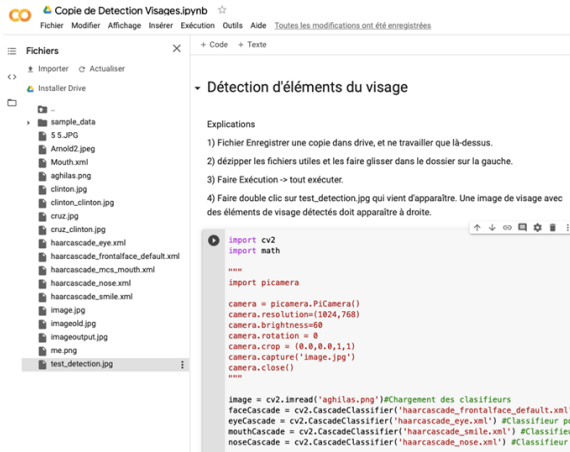


Fig. 11. Illustration of setup 1: Remote execution of a Python Program

### C. Academic impact of the classroom lesson and practicals

The number of students  $N_s$  who participated is  $N_s = 23$ . Their satisfaction was rated, as in section II, with four possible answers from No (1) to Yes (4). A summary of the detailed evaluation, which was done with paper forms, is available at [12]. Sixteen questions were asked to the students. The mean value of all answers is 3.1. Here are four of the questions that were asked to the students (questions 5, 6, 11, 16 in [12]):

- 1) The teacher favors questions during the course ?
- 2) The material used during the course facilitates your learning ?
- 3) This course helps you in defining your professional objectives ?
- 4) This course helps you in consolidating your abilities to work in team ?

In Table II we provide the distribution of the answers to these four questions.

Question index	1	2	3	4
1	0	4	36	59
2	5	10	50	35
3	6	25	37	31
4	11	16	16	56

TABLE II

EVALUATION OF THE CLASSROOM LESSON AND PRACTICALS:  
DISTRIBUTION OF THE ANSWERS TO QUESTIONS 1 TO 4, IN %.

We can notice that the students are interested about the materials presented during the course. One of the students was accepted to perform a trainee with the industrial partner: this course really helps the students defining their professional goals. The students also made some free comments. About the aspects of the course that 'could be improved', they think 'the computers could be a bit faster' and they 'would like to have more time to work on the setups'. About the good aspects of the course: "discover the interest of Raspberry Pi and Arduino", "purpose of the practicals, lessons", "Teachers, practicals", and "value some theoretical side-courses".

## IV. CONCLUSION

A robot produced by an industrial partner is used as a basis for two courses: a project course and a classroom lesson combined with practicals.

In the project course the students come from different Masters. The students have learnt how to control the head and the motor of the robot remotely; they have performed a preliminary study involving Intel cameras with a view to improve the navigation autonomy of the robot. Most of the students have shown their motivation in this course and answered either 'rather yes' or 'yes' to questions rating their level of satisfaction.

In the course which combines a lesson and practicals, the students first have to propose a strategy to improve the autonomy of the robot; then, they learn to know the robot with a series of practicals. Another future goal could consist in studying the battery and the power supply of the robot.

## REFERENCES

- [1] M. M. Alhababha, O. H. A. Mahfoodh, A. Pandian, Y. M. Mohammad, E. W. Ahmed, A. Albdour, and H. Al Bazar, "Check this word out! exploring the factors that affect students' vocabulary learning using smartphones via partial least squares," *Education Research International*, vol. 2014, Article ID 672031, 2014.
- [2] A. T. Avancena, A. Nishihara, and C. Kondo, "Developing an algorithm learning tool for high school introductory computer science," *Education Research International*, vol. 2015, Article ID 840217, 11 pages, 2015.
- [3] R. A. Butt and S. M. U. Ali, "Semantic mapping and motion planning with turtlebot roomba," *IOP Conference Series: Materials Science and Engineering*, vol. 51, p. 012024, dec 2013.
- [4] R. Ghanaatian, O. Afisiadis, M. Cotting, and A. Burg, "Lora digital receiver analysis and implementation," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 1498–1502.
- [5] H. Hua and Z. Zhang, "Application of artificial intelligence technology in short-range logistics drones," in *2019 8th International Symposium on Next Generation Electronics (ISNE)*, 2019, pp. 1–4.
- [6] Intel, "Documentation on intel cameras," <https://dev.intelrealsense.com/docs/datasheets>, 2019.
- [7] —, "Extensions for intel cameras," <https://github.com/IntelRealSense/librealsense/releases/tag/v2.31.0>, 2019.
- [8] —, "Intel cameras sdk," <https://dev.intelrealsense.com/docs/installing-intel-realsense-sdk-20-for-windows>, 2019.
- [9] —, "Sample codes for intel cameras," <https://dev.intelrealsense.com/docs/code-samples>, 2019.
- [10] D. Loukatos and K. G. Arvanitis, "Extending smart phone based techniques to provide ai flavored interaction with diy robots, over wi-fi and lora interfaces," *Education Sciences*, vol. 9, no. 3, p. 224, 2019.
- [11] J. Marot, "Course evaluation: artificial intelligence projects," <https://www.fresnel.fr/perso/marot/Documents/Enseignements/Evaluation/ReportQuestionnaireAI.pdf>, 2020.
- [12] —, "Course evaluation: classroom lessons and practicals," <https://www.fresnel.fr/perso/marot/Documents/Enseignements/Evaluation/EvaluationClassroomCourse.pdf>, 2020.
- [13] J. Marot and S. Bourennane, "Raspberry pi for image processing education," in *Signal Processing Conference (EUSIPCO), 2017 25th European*. IEEE, 2017, pp. 2364–2366.
- [14] Odroid, "Odroid single-board computers," <https://magazine.odroid.com/wp-content/uploads/odroid-xu4-user-manual.pdf>, 2019.
- [15] OpenCV, "Sample codes for object recognition with intel cameras," <https://github.com/opencv/opencv/blob/3.4.0/samples/dnn/>, 2019.
- [16] t. OpenCV, "Open computer vision (opencv)," <http://docs.opencv.org/doc/userguide>, 2018.
- [17] A. Robotics, "Robot ubbo maker," <https://www.axyn.fr/ubbo-maker/>.
- [18] I. Sight Machine, "Simple computer vision (simple cv python)," <http://simplecv.org>, 2018.