UNDER CONTINUOUS DEVELOPMENT

Let's plaiy! - AI at Schools (IP 17, WS 2021)

Design and create teaching material (brainware, software and hardware) for AI / ML / DL / CV in schools.



Powered by NVIDIA Jetson Nano

Introduction

Artificial intelligence is a prevailing technology entails both opportunities and risks to society and the environment. It is crucial for students to acquire these competences in school, enabling them to master the technology at an early age. This allows them to avoid being dominated by AI, to de-mystify AI and to develop the ability for critical reflection in assessing opportunities, applicability and limits of AI for problem solving.

The usability of many AI toolkits is continuously improving. They require less and less expert knowledge to utilise them effectively. This simplification can be taken further. The material we develop should be suited for children from 9 - 11 years. We have to hide the complexity of AI in the background and provide easy to use programming and user interfaces as frontends. An option we will investigate, is to use the open source graphical programming languages SNAP!, similar to Scratch, and MicroBlocks to control AI enabled embedded computers.

The training material to be created focuses on education for sustainable development and other applications oriented towards the common good. Concrete examples are biodiversity monitoring by detecting plant and animal species with robot-borne computer vision and deep learning.

We will develop a course curriculum and tutorial material (brainware), as well as hardware and software for AI-enabled small wheeled robots and tiny DIY drones to be used in education. The open course-ware will be provided and promoted appropriately. The ultimate goal is to design and implement course materials (brainware, hardware, software) for teaching AI in schools.

COURSE DESCRIPTION

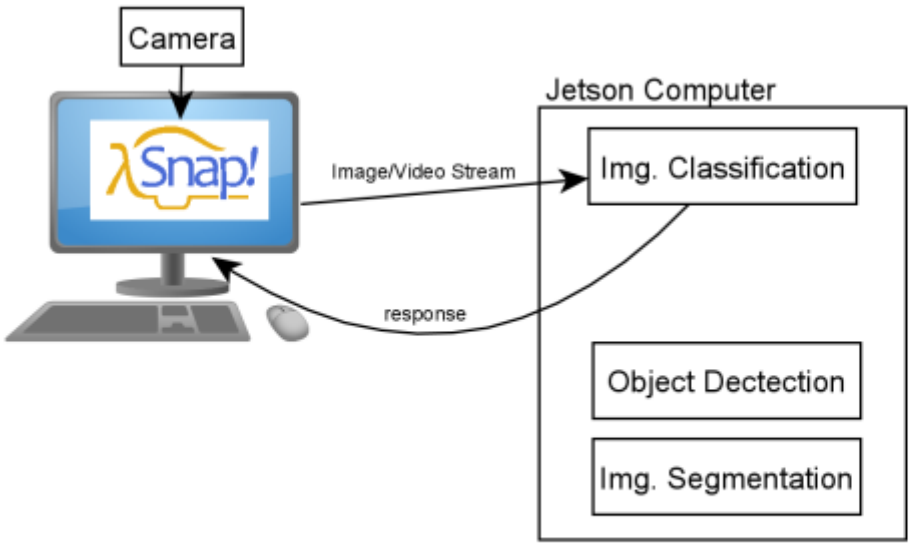
Inspiration

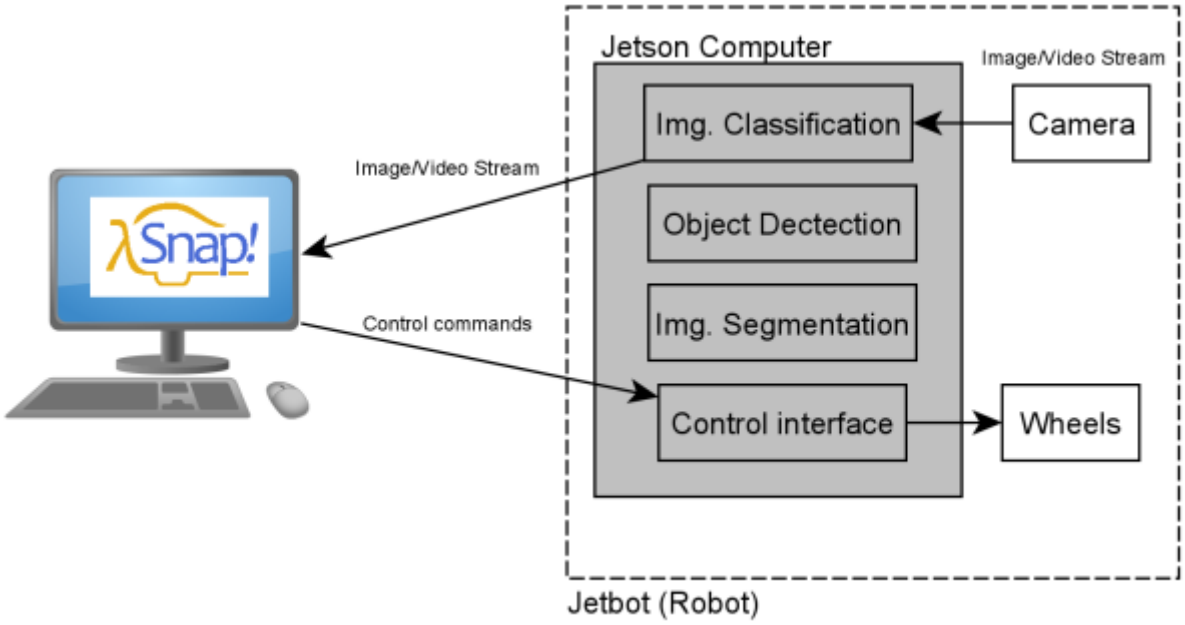
	
1. Introduction Video by Harley Lara, based on the Course Description	**2. SNAP** Jens Mönig demonstrates a simple object classification algorithm for single stroke drawings.
	
3. NVIDIA Jetson Nano Object Detection Inference, Dustin Franklin demonstrates detectnet on Jetson.	**4. Jetbot, HSRW Design**, designed by Harley Lara
	
5. Micro:bit Robot Car Keyestudio Micro:bit Mini Smart Robot Car	**6. MicroBlocks** for Micro:bit

	
7. Educational Drone 1, Parrot Mambo (discontinued)	**8. Educational Drone 2**, Ryze Tello EDU

Distributed Architecture

The system is made of several interacting components which can be distributed across the hardware in different manners.





Schedule

Session	Date	Location	Topics	Intended Session Outcome	Lab Exercise	Session Log	Assignments
1	30.09.2021	IoT Lab (02 02 510)	Introduction	Jetsons assembled	Jetson Nano Assembly	2021-09-30	
2	07.10.2021	IoT Lab (02 02 510)	Software Installation on Jetsons	Systems ready to use	Basic Setup of NVIDIA Jetson Nano	2021-10-07	Learn Linux

Session	Date	Location	Topics	Intended Session Outcome	Lab Exercise	Session Log	Assignments
3	14.10.2021	IoT Lab (02 02 510)	Playing with object detection	Practical experience how to run the OD docker container, insight into strengths and limitations of OD	Prepare the Jetsons to run object detection (detectnet) with one of the CSI cams: Hello Object Detection	2021-10-14	Play with SNAP!
4	21.10.2021	IoT Lab (02 02 510)	Hello API from Snap!	Practical experience how to get data from the EOLab Wheather Station API and display information into Snap!	Hello API with Snap!	2021-10-21	Develop proposals on how to communicate Snap! with a backend for image processing
5	28.10.2021	IoT Lab (02 02 510)	Snap! and backend communication	Understand the communication system between Snap! and the Jetson Nano and communicate with the test server	Send base64-encoded image from Snap! to the Flask server running on the Jetson Nano and get the server's reply	2021-10-28	- Setup the Flask server on your own Jetson Nano - Communicating a Snap! project with the Flask server - Create a Demo in Snap! that could be interesting to use as didactic material, each student must present his or her demo in the next session.
6	04.11.2021	IoT Lab (02 02 510)	Presentation of project ideas	Identify attractive project proposals to be implemented as didactic demonstrators in the learning material, in addition to providing support and guidelines to particular problems faced by each group.	Each group presents progress in software setup in the Backend and communication with Snap.	2021-11-04	1) Install Ilgar's and Harley's scripts on frontend (SNAP!) and backend (flask, python). Send images from SNAP! to the image classifier and receive the results. 2) Create a presentation on your ideas of how to realize the training. Several software architectures are possible. **Let's plaiy! Git repository**
7	11.11.2021	IoT Lab (02 02 510)					

Session	Date	Location	Topics	Intended Session Outcome	Lab Exercise	Session Log	Assignments
8	18.11.2021	IoT Lab (02 02 510)				2021-11-18	1) **In group**: create a didactic project using Snap! and Nvidia AI models and **document it** on the **Wiki or Moodle**. The project should be replicable by children, and the material should serve as guide for the project. In the next session they should **present the project and the documentation**. 2) **In group**: Propose using a diagram an architecture/workflow for the **training task (as detailed as possible)**, it should contain information flow, communication methods between components, interfaces etc. How can we manage the training? How can children take own photos, label them and use them for transfer learning?
9	25.11.2021	IoT Lab (02 02 510)					
10	02.12.2021	IoT Lab (02 02 510)				2021-12-02	
11	09.12.2021						
12	16.12.2021						
—	~~23.12.2021~~						
—	~~30.12.2021~~						
13	06.01.2022						
14	13.01.2022						
15	20.01.2022	Foyer of Audimax	Poster Session				

Work Packages

Software AI

- Implement Object detection with DetectNet
- Create a wrapper to communicate the neural network with Snap!

Requirements:

- Modular Design
- Good coding practices
- Use standard communication protocols

Software Snap

- Interfacing Snap (Front-end) with Jetson Nano (back-end)
  - WebSocket API ?
  - Web Serial API ?
  - Web USB API ?
  - Custom module inside Snap! ?

Requirements:

- Modular Design
- Good coding practices
- Use standard communication protocols

Hardware Jetbots

- Assemble prototype
- Create assembly instructions
- Electronics wiring diagrams

Requirements:

- Easy assembly
- Assembly using common tools
- No specialized machinery
- Modular design with the possibility to remove/place the Jetson Nano easily

Deliverables

Main goal: Develop, design and implement teaching materials on artificial intelligence (AI) for schools (primary schools, age 9-11 years).

- SNAP! course for kids

- Jetbot assembly tutorial
- Replicable projects using SNAP! as frontend:
  - Image classification
  - Predefined object detection with camera: (or rtp, rtsp, rtmp) with ssd-mobilenet
  - Control the Jetbot robot
  - Training convolutional neural network (CNN) for detection of new objects



The complexity of all projects should be hidden, so all projects should be easy to implement using SNAP! as the interface.

Deliverables, requirements

- **Learning path (easy to follow)**: The teaching material should be easy to follow, with a defined structure that connects each of the topics in a way that is easy to absorb. The material can even have a story telling that proposes solutions to current problems such as global warming, garbage in the oceans, deforestation and more.
- **Content**: The content of the material may be supported by different multimedia means such as videos, images, audio, etc. The way of presenting the content must be precise and easy to understand.
- **Interactive**: The content must be interactive, with gamification strategies that encourage curiosity to continue learning.
- **Learning experience**: The learning experience must be supported by proposed mini-projects, challenges, examples and a final project that unifies all that has been learned.
- **Appropriate use of licenses**: The teaching material must be open and accessible to everybody under the license CC Attribution-Noncommercial-Share Alike 4.0 International so it is important to make use of resources (videos, audios, images and others) that have a license that allows copying, redistribution, modification and transformation, as far as possible *you should generate your own content under the same license of the course*.

Documentation

- **Student Documentation Space**

Internal Links

- **TEACHERS' TO DO**
- Session Logs

* linux

External Resources

1. It’s not Magic After All - Machine Learning in Snap! using Reinforcement Learning, Sven Jatzlau et al.
2. I Programming by Children using Snap! Block Programming in a Developing Country, by Ken Kahn, Rani Megasari, Erna Piantari, and Enjun Junaeti
3. Proceedings of the 2020 Constructionism Conference
4. Hyper Blocks in Snap! v6
5. External Link

From:
https://student-wiki.eolab.de/ - **HSRW EOLab Students Wiki**

Permanent link:
https://student-wiki.eolab.de/doku.php?id=ip:ws2021:lets_plaiy:start&rev=1643733393

Last update: **2023/01/05 14:38**