

IDS - Autumn 2021 -Hand-in #2

Create a an interactive system or interface with machine learning at it's core:

- You're welcome to use Teachable Machines, [yolo](#), [Google's MediaPipe](#), [openCV's object-detection](#), [wekinator](#), speech recognition, self-trained models or any pre-trained model as your base or focus.
- The program has to save and/or load from something disk. This could be [logging](#), saving images, pickled machine learning models, JSON files with user settings etc.

Formal requirements

- Work in groups of 2-6. (You can decide to do same groups as last hand-in)
- **Important:** you must upload both hand-in 1 and hand-in 2 as your final hand-in.
 - Create a directory with a "part 1" and "part 2".
- For the second part, write up to 6 pages (excluding screenshots!) describing your program, your code, your use of data structures, design, the machine learning aspect and what ever else you see fit.
- Upload your writing as .pdf and your code in a zip file. Ideally also include a link if you use github.
 - Make sure that you include a readme file showing how to run the program (including pip modules needed or any setup required).
- A short (max 1.5 min.) video showing the program (from hand-in 2) in use.

Ideas

- Create software that allows for mouse-free control of the computer using [pyautogui](#) and [MediaPipe](#).
- Train [TeachableMachines](#) to recognise objects, take X number of photos of different ingredients and get a recipe of a recipe using these ingredients with [with a recipe API](#). Make it usable and interactive!
- Create a a system where two clients can send emojis back and forth using object and/or facial expression recognition using UDP packets.
- Build a dataset generator, where e.g. poses or images are saved and labelled in an interface. Optionally do some kind of machine learning on these datasets. (pretty advanced!)
- Create a program allowing you to draw on the screen without the mouse!
- The game Pong where you control using face tracking.
- Creating a voice-assistant connected to an API. "Hey IDS, what's the weather?"-style.

Ideas are allowed to be absurd and experimental, but prioritise features and approaches that allow for plenty of programming!

Additional info:

- You are welcome to write in C++ (ESP32), Javascript or Python. Utilize elements from this course!
- You can consider outputting UDP messages and training models using [wekinator](#).
- You are more than welcome to combine several of the simpler ideas from above.
- An ambitious scope is rewarded!
- It's alright if some elements do not work. Be sure to discuss and document why and how it doesn't.

Some elements of a good or excellent hand-in is:

- Usage and mastering of the techniques and data structures presented during the course.
- Describe the theory or mechanics behind the more advanced stuff you do.
- Well-written and well-organized code.
- Demonstration and thoughts of how you tested your program.
- Thorough descriptions of the program, your code, your use of data types, your thought processes and reflections/discussions on extendability, difficulties and your design process.
- Thoughts on your use of machine learning and how it affected the design process.
- Nice coding style! Well formatted, apt use of functions.

Additional links, ideas and inspiration

- [Simple hand gesture recognition](#)
- [ml4a guides](#)
- [TensorFlow models](#)