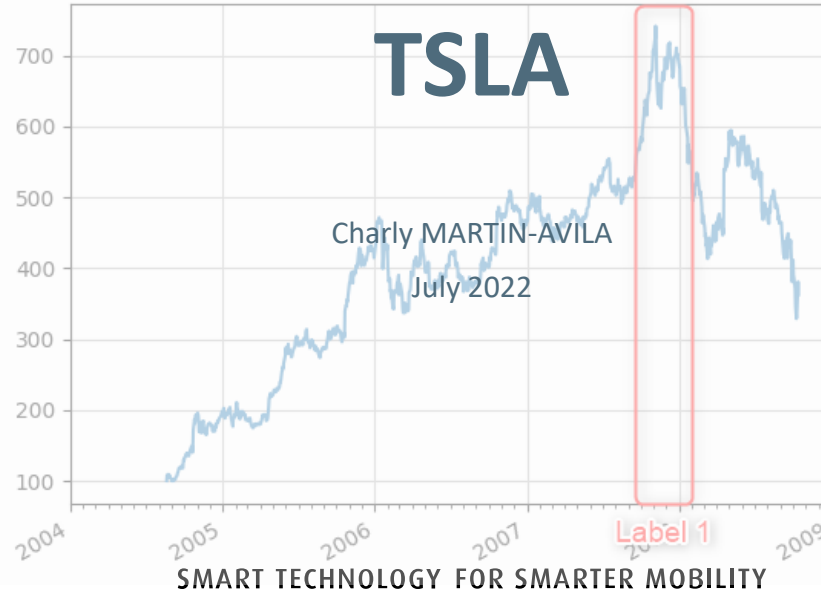




Project Restitution Time Series LAbelling



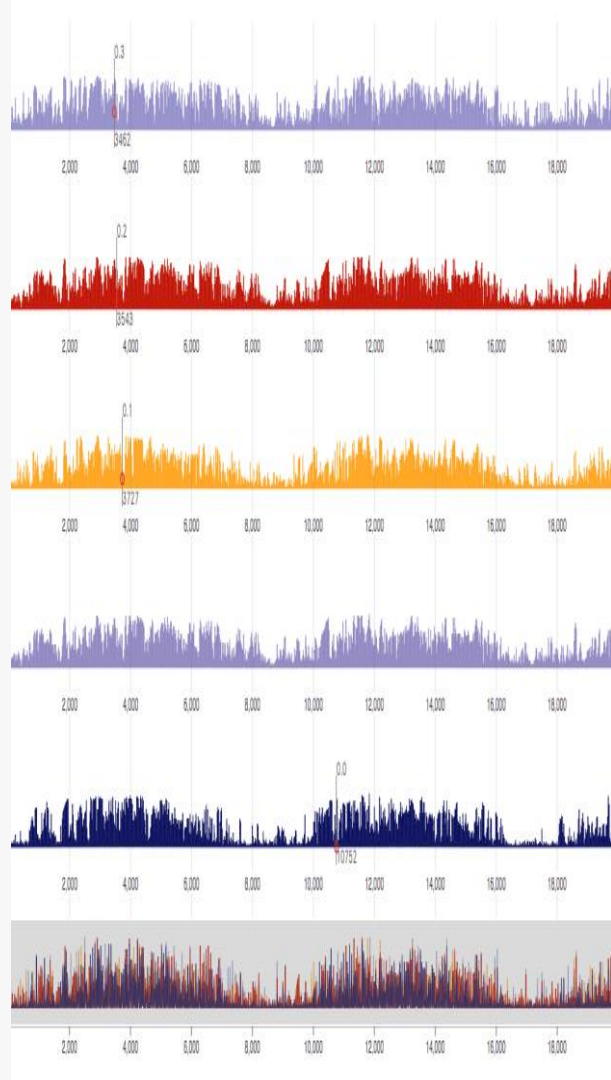


Table of contents

1	Project Description	1
2	The app	12
3	Conclusion	22
4	Perspectives	33

Project Description



Labelling and AI

In any AI project, data issues are some of the sticking points.

Sometimes, data needed for a project may not exist at all. What makes data suitable or unsuitable for feeding into the computers?

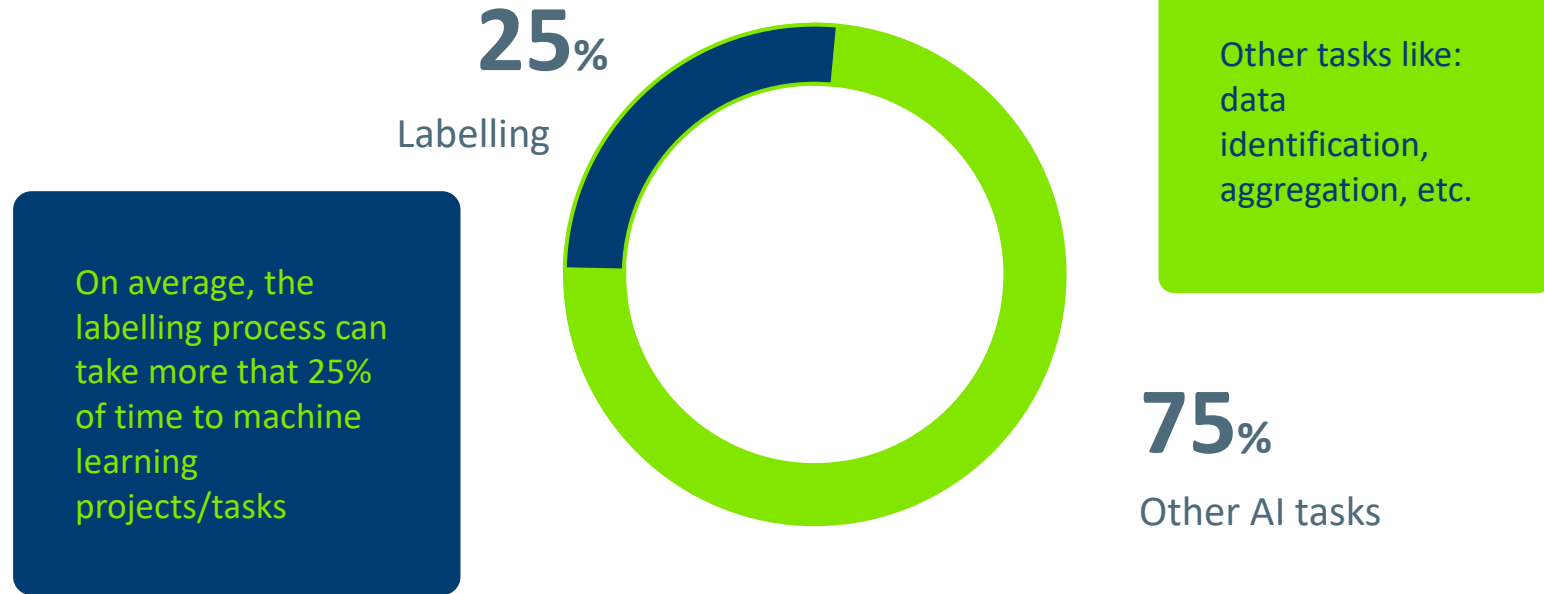
The answer lies in data labeling.



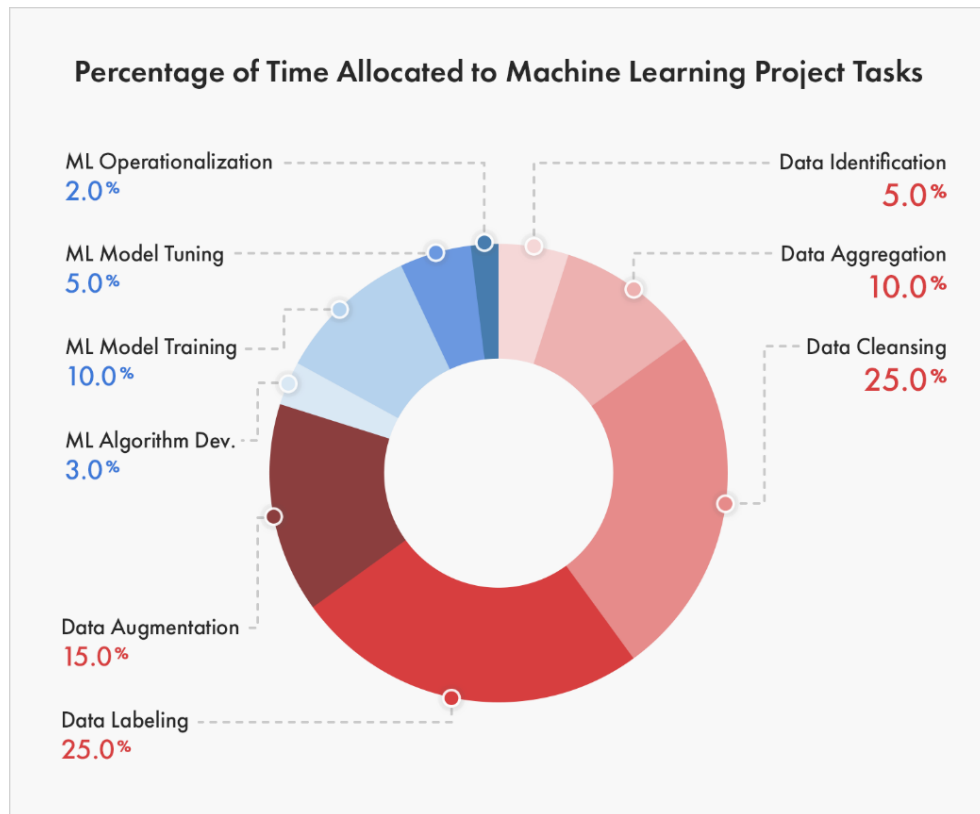
Labelling

Process of identifying raw data (images, text files, videos, etc.) and adding one or more meaningful and informative labels to provide context so that a machine learning model can learn from it.

Time allocated to AI Labelling Task



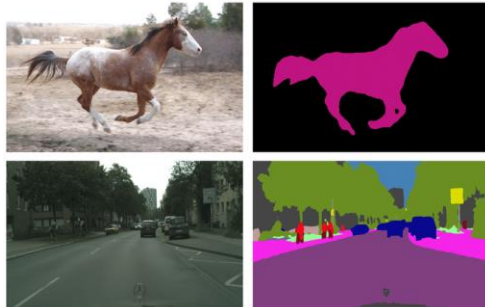
Time allocated to AI Labelling Task



What are the most important labelings types ?

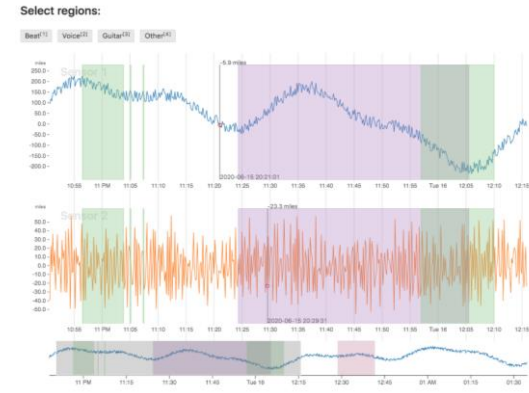
Computer Vision (image segmentation)

It comes from modelling image processing using the techniques of machine learning. It applies machine learning to recognise patterns for interpretation of images.



Time series labelling

Labelling process with data (sensors, events, signals ..)
Tagging ML data by events happening in the time



Labeling Process : Our Target



Manually

Our use case demand a visualisation video frame by frame in order to assign a label



Accelerate

The labelling process



Expensive

The labelling utils are (really) rare. And the most part, the doesn't support video files (signals/data only).



Automate

The labelling process through the util

Labeling Process: ressources & costs

Our use case is based on the labelling of time series data.

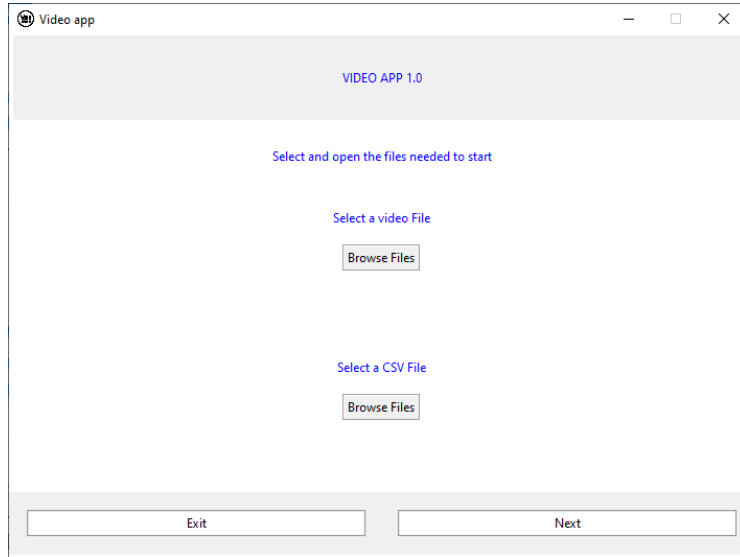
Most labelling tools are not made to display a video synchronised with
the flux of data

The App

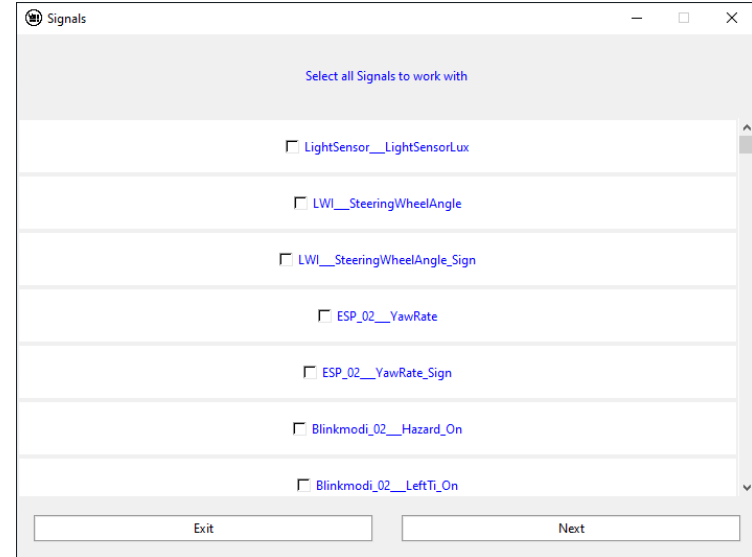
Project tree

```
video-desktop-app
├── documentation
│   ├── browserWindow.md
│   ├── instructions.md
│   ├── labelsWindow.md
│   ├── mainWindow.md
│   └── signalsWindow.md
├── extras
│   ├── images
│   │   ├── icon.png
│   │   ├── labels-example.PNG
│   │   ├── labelswindow.PNG
│   │   ├── main-window-beta-2.PNG
│   │   ├── main-window-beta.PNG
│   │   └── button-players.jpg
│   └── src
│       ├── core
│       │   ├── connexion.py
│       │   ├── label.py
│       │   ├── model.py
│       │   ├── signal.py
│       │   └── video.py
│       └── ui
│           ├── assets
│           │   ├── labelUI.py
│           │   ├── signalUI.py
│           │   └── videoUI.py
│           ├── browserFileWindow.py
│           ├── labelsWindow.py
│           └── mainWindow.py
```

Signal & browser windows



The Browser window allows to select the video and csv/excel file in order to read them



In the signal window , all signals from the csv/file will be displayed to chose the ones to work with

Label windows

The screenshot shows a window titled "Video app" with a header "Enter all labels needed". Below the header is a table with three columns: "id", "Name", and "Color". The "id" column has a text input field. The "Name" column has a text input field. The "Color" column has a purple color selection box. To the right of the "Color" column is a "Delete label" button. Below the table is a "Change" button. At the bottom of the window is an "Add a new Label" button. At the very bottom are "Exit" and "Next" buttons.

id	Name	Color
<input type="text"/>	<input type="text"/>	<div style="background-color: purple; width: 20px; height: 20px;"></div>

Change

Delete label

Add a new Label

Exit Next

On the label Window, you enter the labels you will work with

The screenshot shows the same "Video app" window, but now it contains three rows of label data. The first row has id "0", name "name1", and a purple color. The second row has id "1", name "name2", and a green color. The third row has id "2", name "name3", and a teal color. Each row has its own "Change" and "Delete label" buttons. The "Add a new Label" button is still at the bottom.

id	Name	Color
0	name1	<div style="background-color: purple; width: 20px; height: 20px;"></div>
1	name2	<div style="background-color: green; width: 20px; height: 20px;"></div>
2	name3	<div style="background-color: teal; width: 20px; height: 20px;"></div>

Change

Delete label

Change

Delete label

Change

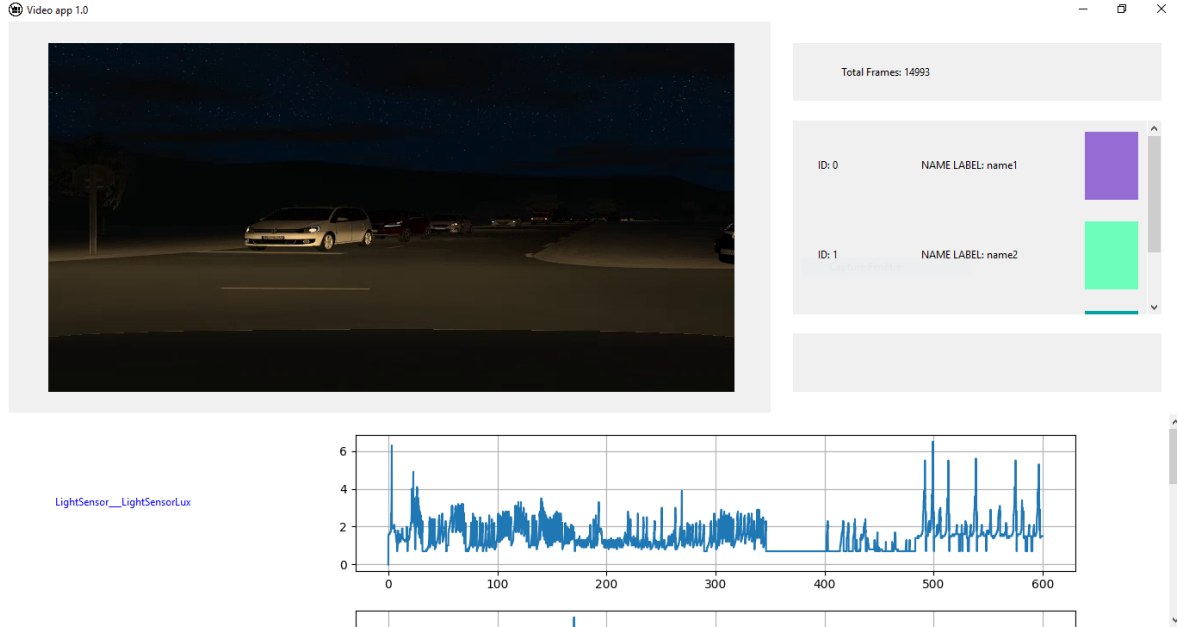
Delete label

Add a new Label

Exit Next

In this example, there are 3 labels with different colors and IDs

Main Window



Work space for
labeling the signals file
(csv)

Choice of libraries



04

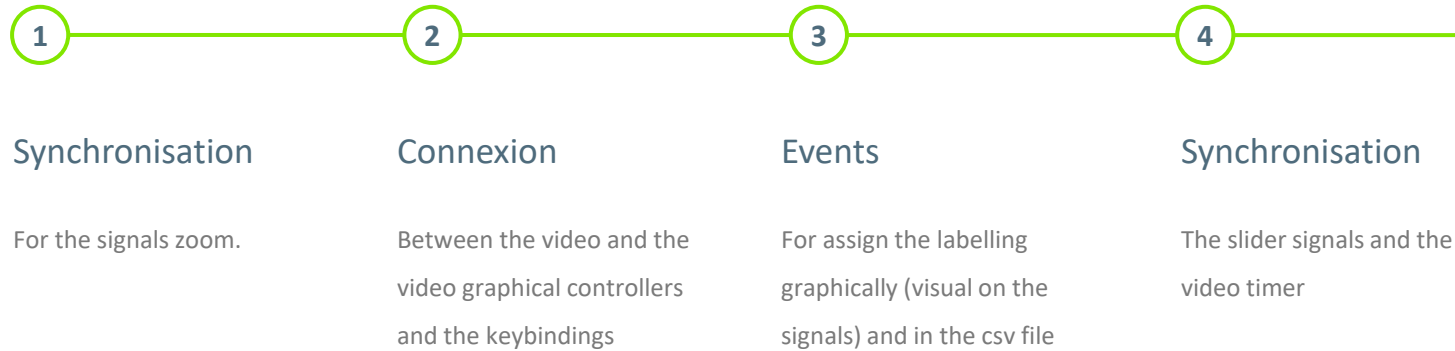
Conclusion



- The labelling process is one of the most important tasks for any AI project
- Time series data are so variant shapes, so we have a specific need to annotate our data for power consumption
- A tool is implemented to accelerate and automate the labelling process
- As a limited period, most of principal components of interface were made, but some features are still remained to do

Perspectives

To do implementations



Ideas implementations

- ➔ An undo/redo buttons for editing the csv file
- ➔ A 'save' option in order to save the current state of the labelling and a 'open file' to take over the labelling edition
- ➔ Import automatically signals preferences for a same file: Not to select all labels multiple times



SMART TECHNOLOGY
FOR SMARTER MOBILITY