The use of artificial intelligence and sensors to monitor and classify objects.

FAAI:
The Future is In Applied Artificial Intelligence

dr Marcin Bernaś

Disclaimer: Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the National Agency (NA). Neither the European Union nor NA can be held responsible for them.

Co-funded by the European Union

Agenda

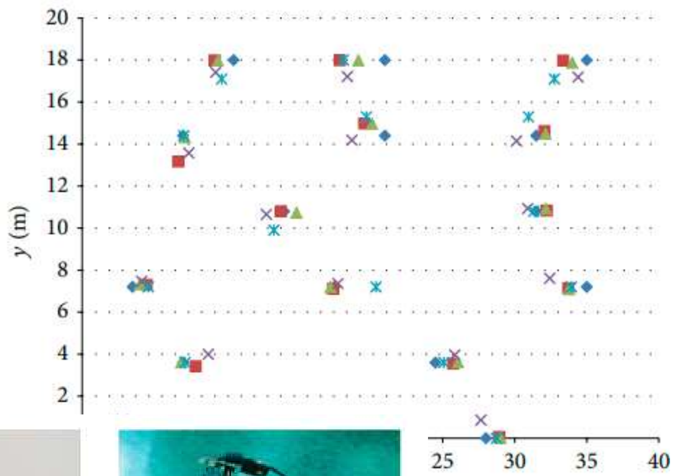
- My projects with AI
- The modern approach
- Scenario 1 – use trained model
- Scenario 2 – create own model

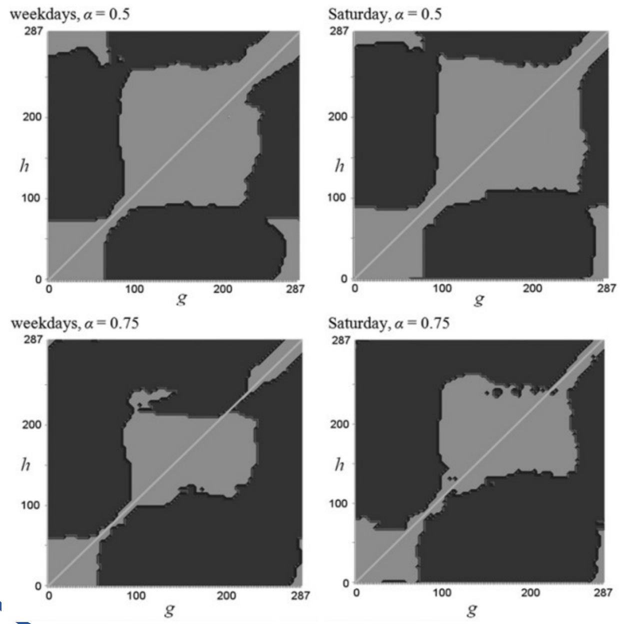
FAAI:2022-1-PL01-KA220-HED-000088359

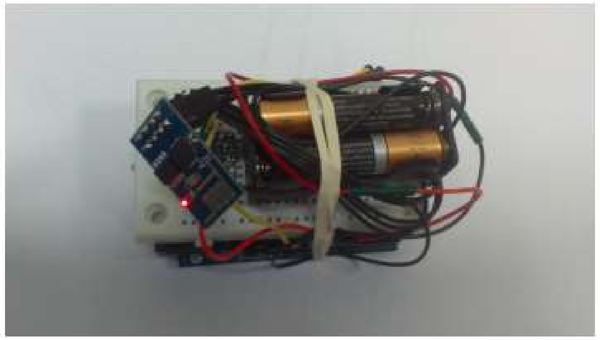
Co-funded by the European Union

Main research area

1. **Mobile and Wireless Sensor Networks** (WSNs).
2. **Machine Learning methods**
3. **Vehicular Networks**

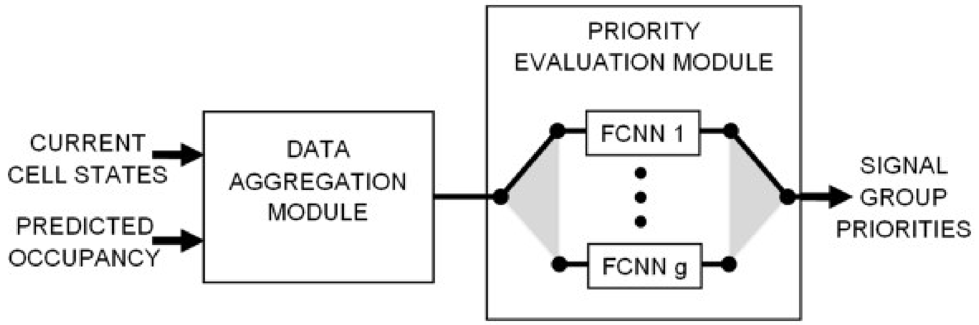






Co-funded by
the European Union

Neuroevolutionary Approach to Controlling Traffic Signals



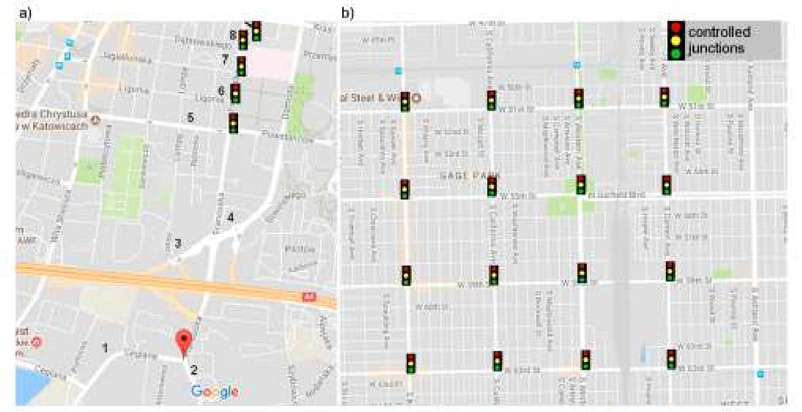
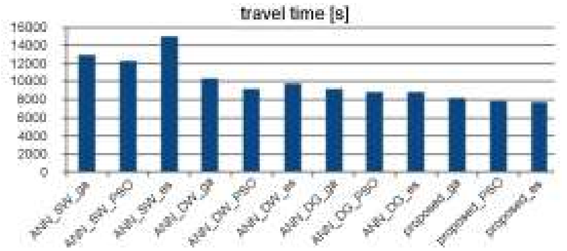
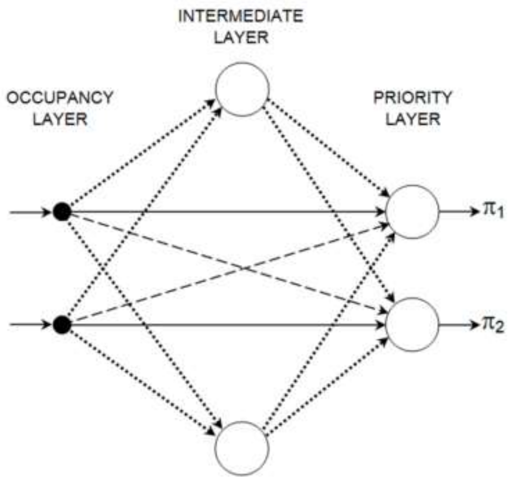


Figure 8. Test scenarios: (a) signalized arterial road; and (b) grid road network (based on google maps: http://maps.google.com).





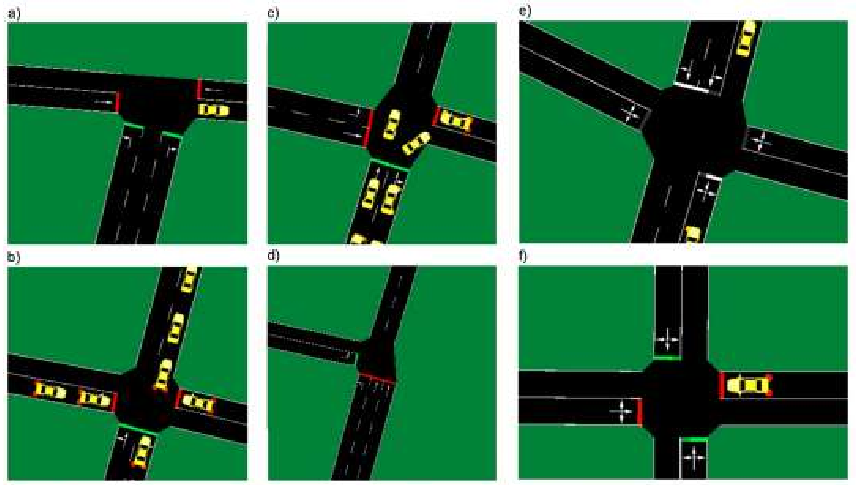
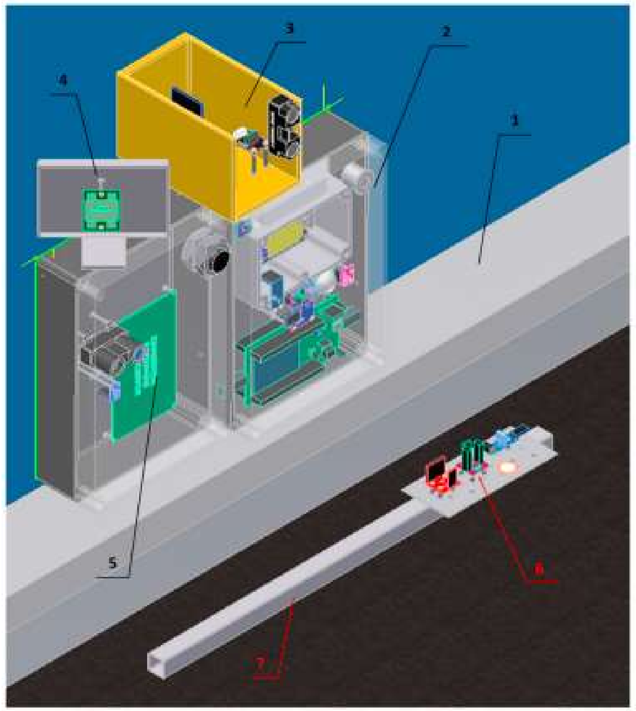


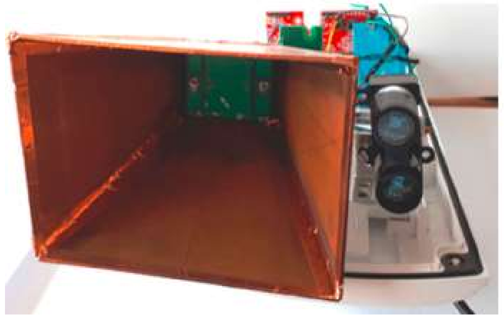
Figure 9. Intersections characteristic for: the first scenario (a–d); and the second scenario (e,f).

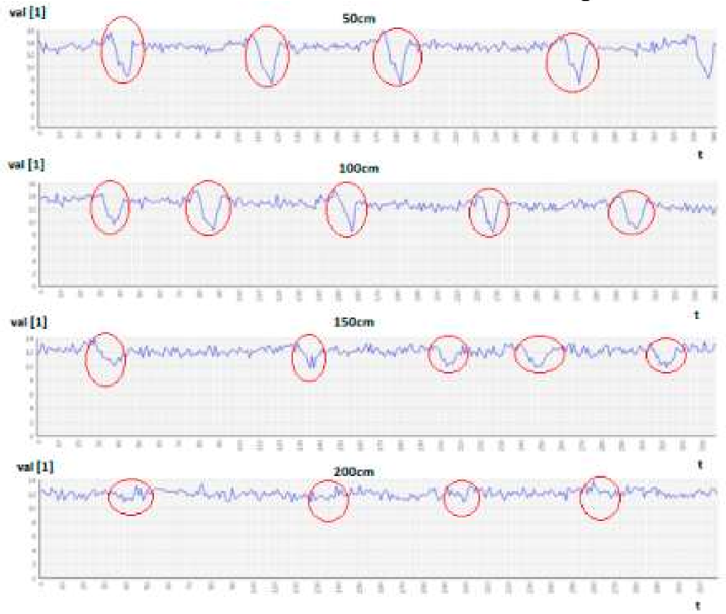
FAAI:2022-1-PL01-KA220-HED-000088359

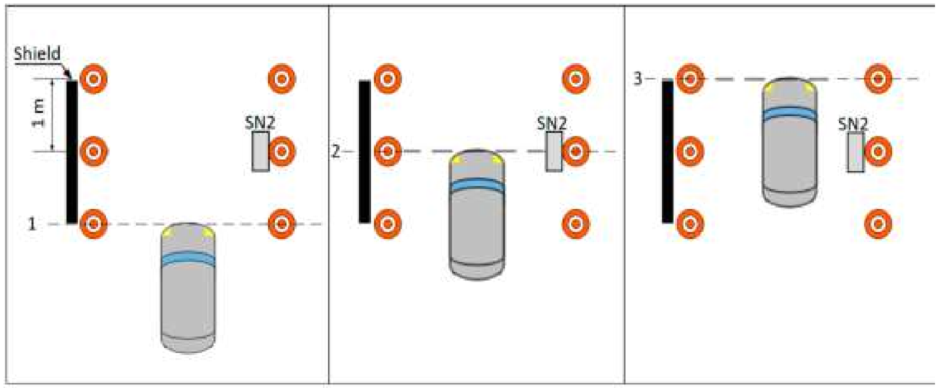
Co-funded by the European Union

Vehicles tracking



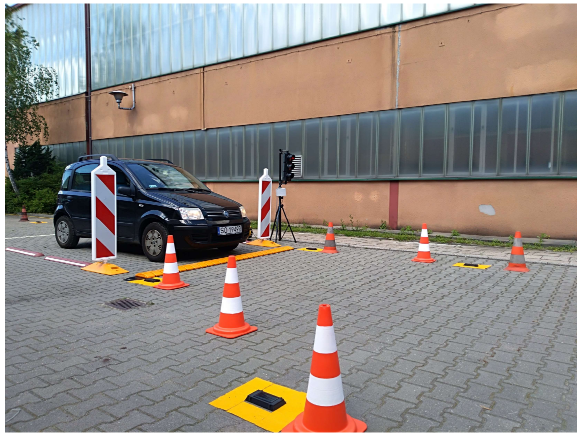


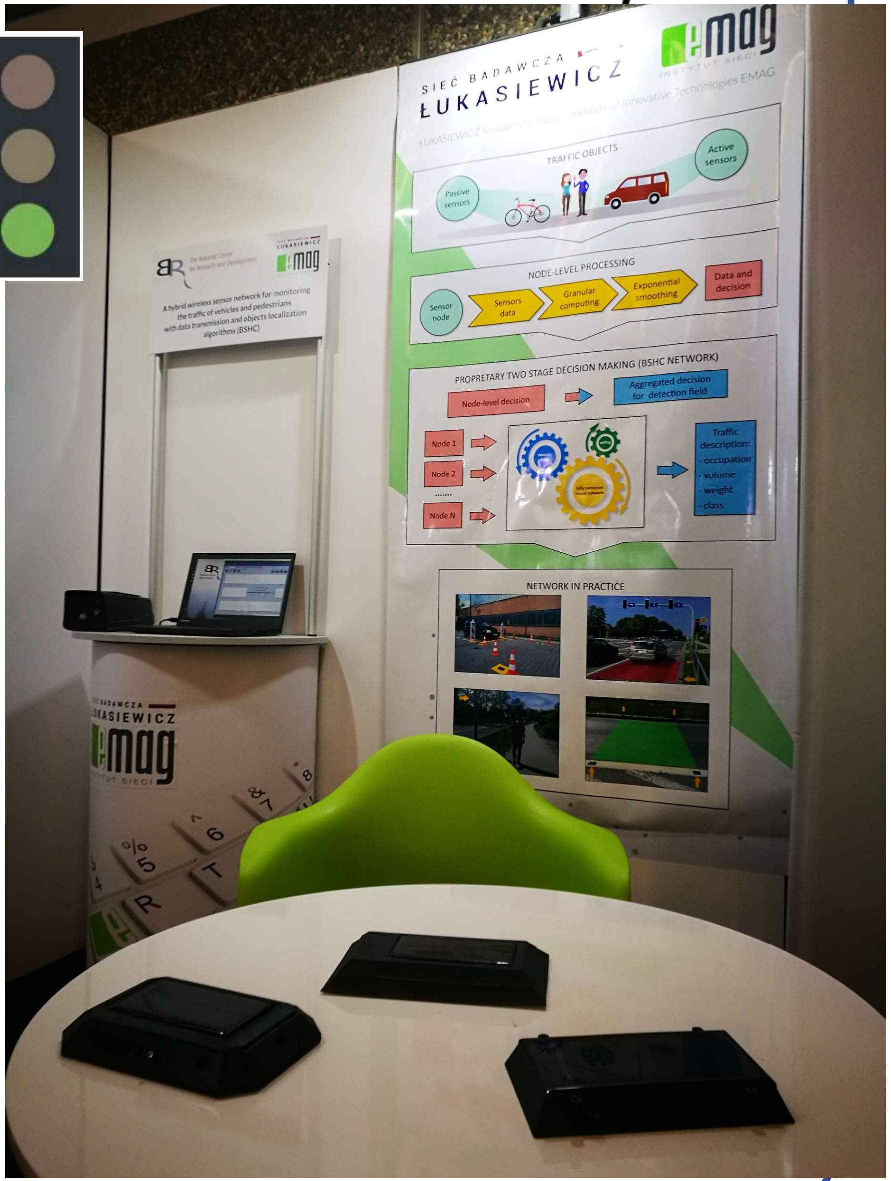


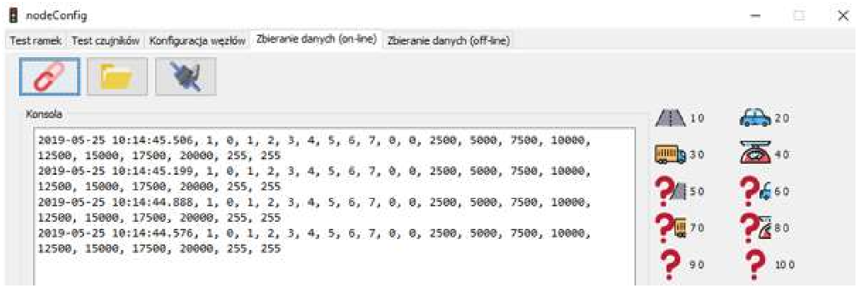


Co-funded by
the European Union

Implementation



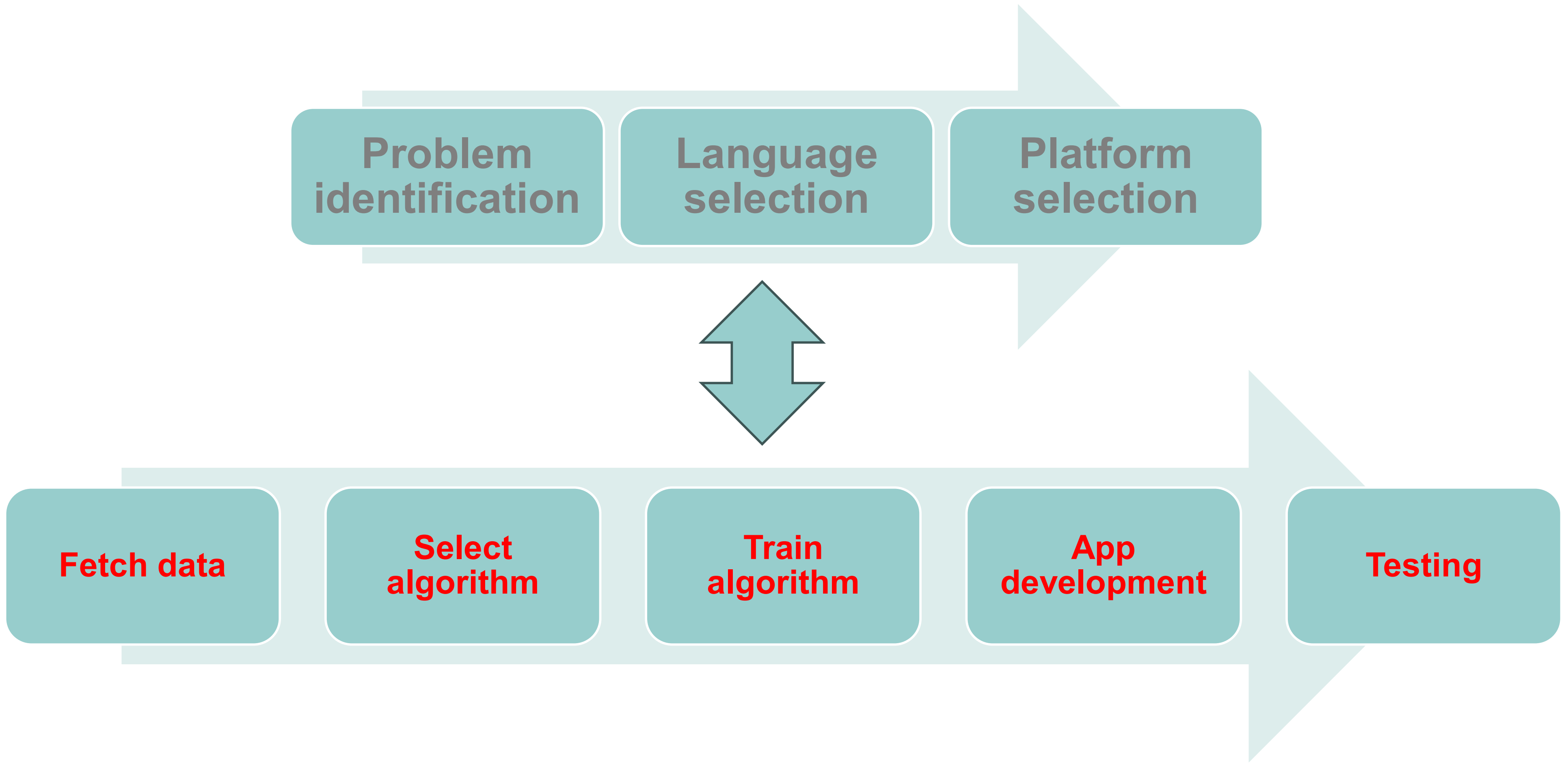




```
2019-05-25 10:14:45.506, 1, 0, 1, 2, 3, 4, 5, 6, 7, 0, 0, 2500, 5000, 7500, 10000,
12500, 15000, 17500, 20000, 255, 255
2019-05-25 10:14:45.199, 1, 0, 1, 2, 3, 4, 5, 6, 7, 0, 0, 2500, 5000, 7500, 10000,
12500, 15000, 17500, 20000, 255, 255
2019-05-25 10:14:44.888, 1, 0, 1, 2, 3, 4, 5, 6, 7, 0, 0, 2500, 5000, 7500, 10000,
12500, 15000, 17500, 20000, 255, 255
2019-05-25 10:14:44.576, 1, 0, 1, 2, 3, 4, 5, 6, 7, 0, 0, 2500, 5000, 7500, 10000,
12500, 15000, 17500, 20000, 255, 255
```

Co-funded by
the European Union

Creation of app with AI



FAAI:2022-1-PL01-KA220-HED-000088359

Co-funded by
the European Union

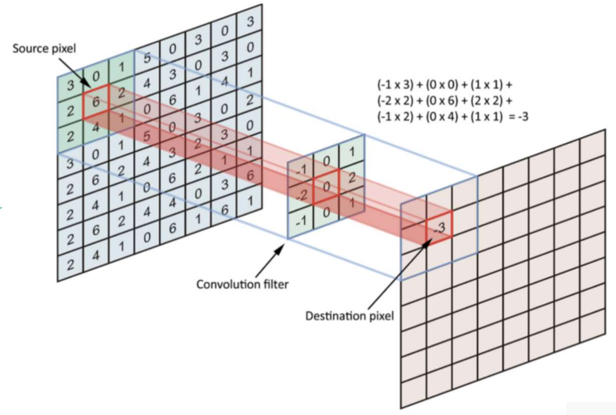
Use case 1 – black box approach

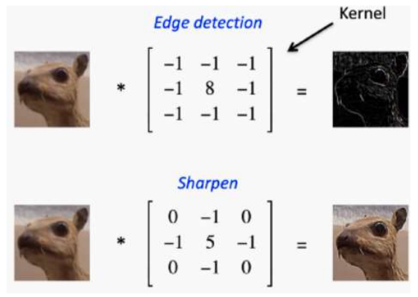
- The gesture tracking can find applications in many fields:
  - using immersive virtual reality
  - accesing user interface elements
  - in healthcare using precise movement monitoring
  - gaming while natural hand interaction
- In this example:
  - module aims to return position of hands and detect set of gestures
  - it will be used to open and close applications.

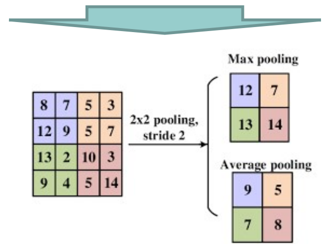
Co-funded by
the European Union

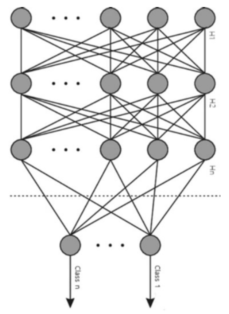
Convolution NN (already done?)

- There are **three types** of layers:
  - convolutional layer (catches features),
  - pooling layer (reduce size),
  - fully connected layer (makes classification).


Source pixel
(-1 x 3) + (0 x 0) + (1 x 1) +
(-2 x 2) + (0 x 6) + (2 x 2) +
(-1 x 2) + (0 x 4) + (1 x 1) = -3
Convolution filter
Destination pixel







Co-funded by the European Union

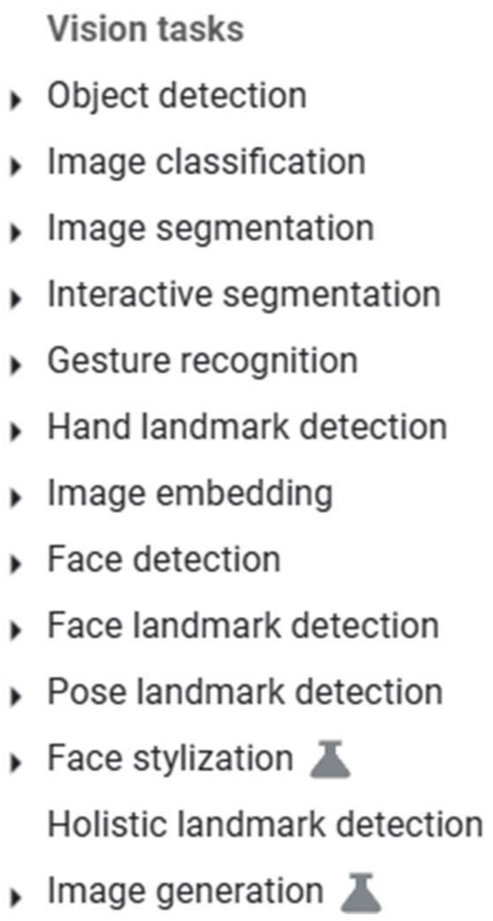
Mediapipe

- Solution simplifying deploying machine learning models on devices, making them production-ready and accessible across various platforms.
- The solution:
  - Contin lightweight models with high accuracy
  - Supports vision, text, and audio processing
  - Is accelerates on both CPU and GPU
  - Accesible on Android, iOS and web.

Online repository:
MediaPipe | Google for Developers

Co-funded by the European Union

Mediapipe

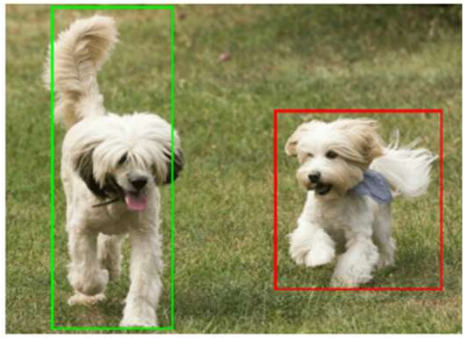


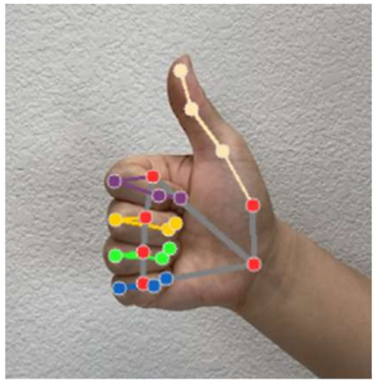
Vision tasks

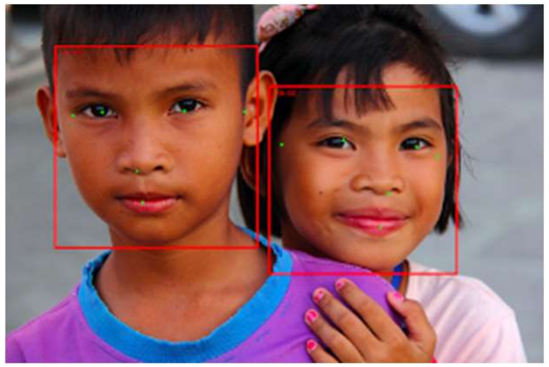
- Object detection
- Image classification
- Image segmentation
- Interactive segmentation
- Gesture recognition
- Hand landmark detection
- Image embedding
- Face detection
- Face landmark detection
- Pose landmark detection
- Face stylization

Holistic landmark detection

- Image generation







[https://developers.google.com/mediapipe/solutions/vision]

Co-funded by the European Union

Models

- Mediapipe offers build in models for image, sound and text processing. In this case the hand tracking will be used thus following models will be implemented:
  - gesture_recognizer – model which tracks pretrained hand gesture – in total 8 gestures are suported
  - hand_landmarker – allows to describe hand as set of descriptors (including each hand bone)

Co-funded by the European Union

Implementation of a model

- To use a AI module the following elements will be implemented:
  - Obtain a data in module
  - Load AI model
  - Process the data
  - Fetch and process the result

- The example will be presented using Python language

Co-funded by
the European Union

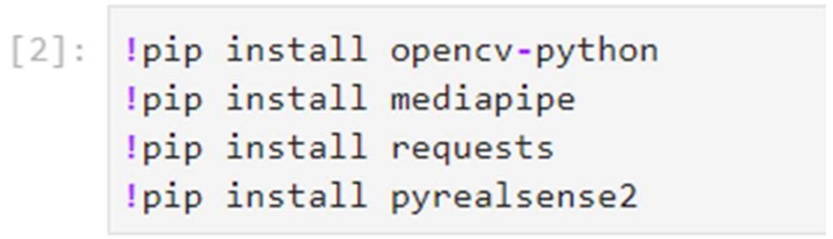
Environment settings

- The script can be applied in environment, where the access to camera and application is possible
- Therefore, the example can be execute using Anaconda environment (access to local computer processes).
- The processing algorithm can also work on servers solutions.

Co-funded by
the European Union

Importing libraries

- The usecase is build based on mediapipe library. However the picture manipulation is based on opencv library. Additionally, suporting libraries were used.

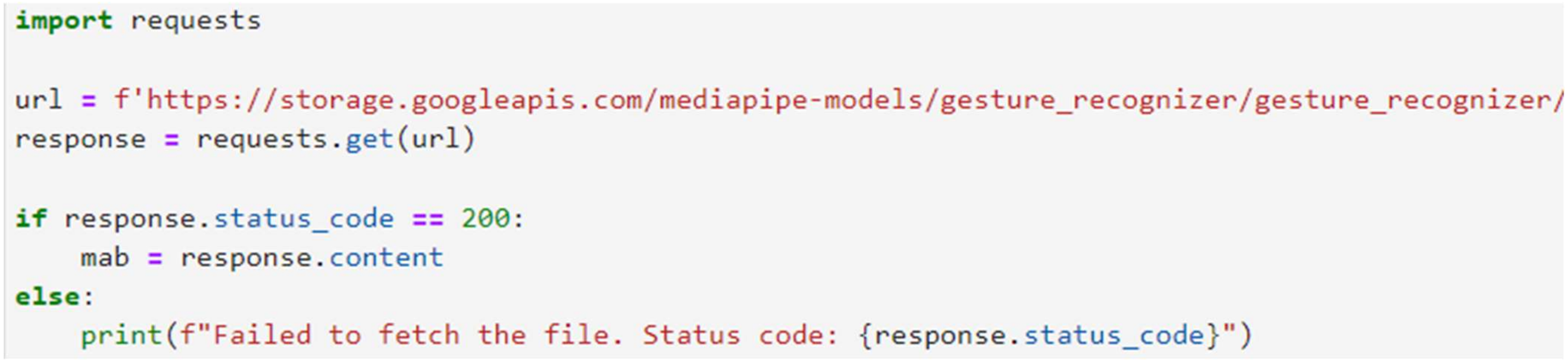


```
[2]: !pip install opencv-python
     !pip install mediapipe
     !pip install requests
     !pip install pyrealsense2
```

Co-funded by the European Union

AI models

- The models for mediapipe are created as routines or tasks that can be easily used after configuration.
- The models in example are downloaded directly from Google server.

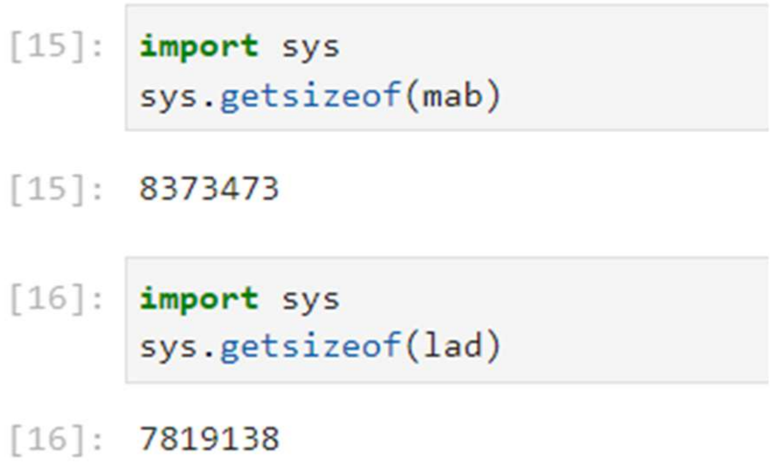


```
import requests

url = f'https://storage.googleapis.com/mediapipe-models/gesture_recognizer/gesture_recognizer/
response = requests.get(url)

if response.status_code == 200:
    mab = response.content
else:
    print(f"Failed to fetch the file. Status code: {response.status_code}")
```

Co-funded by
the European Union

Model verification

- The should be verify by checking its size of check sum
- Due to used download method the size verification is sufficient

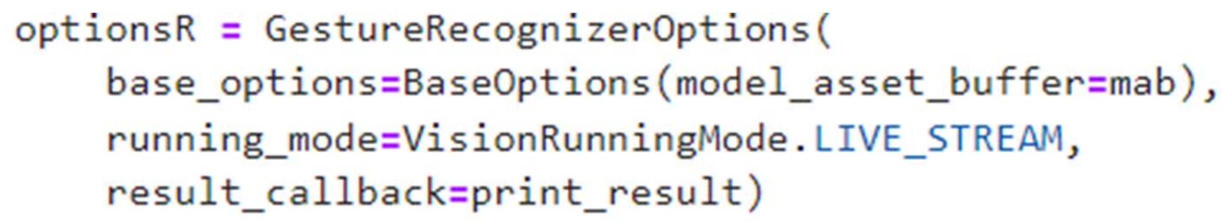


```
[15]: import sys
      sys.getsizeof(mab)

[15]: 8373473

[16]: import sys
      sys.getsizeof(lad)

[16]: 7819138
```

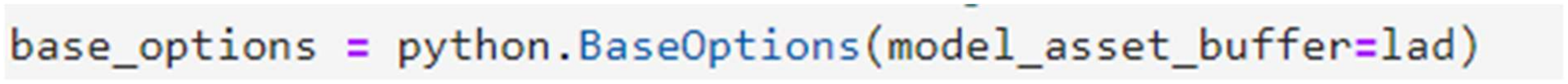
Co-funded by the European Union

Suport functions

- To create the recognizer using mediapipe the model should be configured.
- The configuration is made by seting the options.



```
optionsR = GestureRecognizerOptions(
    base_options=BaseOptions(model_asset_buffer=mab),
    running_mode=VisionRunningMode.LIVE_STREAM,
    result_callback=print_result)
```



```
base_options = python.BaseOptions(model_asset_buffer=lad)
```

Co-funded by
the European Union

Creating AI module

- AI module follows the simple routine for each image obtained from camera:
  - Send image to hand recognition routine
  - Send image to gesture recognition routine
  - Obtain results from routines and display it as image

Co-funded by
the European Union

Procesing using AI models

- Gesture recognition:

```
# Convert RGB image to MediaPipe Image format
mp_image = mp.Image(image_format=mp.ImageFormat.SRGB, data=frame)
# gesture recornition routine using AI model
if mp_image:
    with GestureRecognizer.create_from_options(optionsR) as recognizer:
        recognizer.recognize_async(mp_image, 50)
```

- Hand tracking:

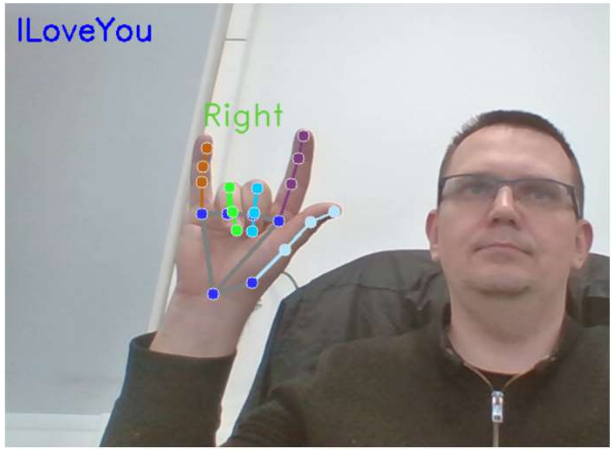
```
#hand tracking using AI model
detection_result = detectorH.detect(mp_image)
```

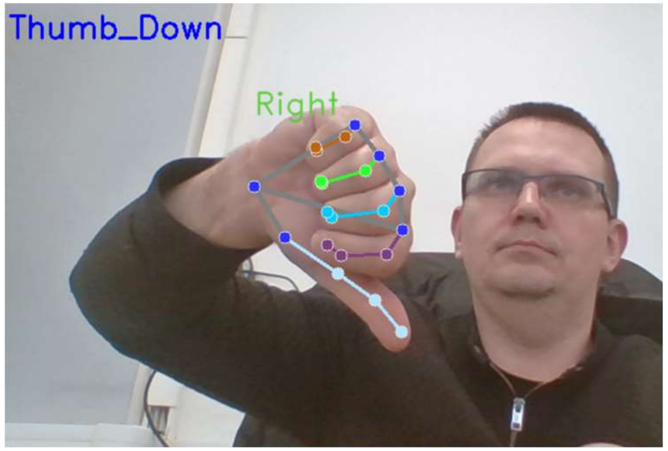
Co-funded by
the European Union

Process a results

- The result as a detection class and set of landmarks can be displayed as an image:

```
# drawing landmarks on detected hand
frame = draw_landmarks_on_image(mp_image.numpy_view(), detection_result)
#conversion between the color palate BGR/RGB
frame=cv2.cvtColor(frame,cv2.COLOR_RGB2BGR)
# description of detected class in image
cv2.putText(frame,detectN, bottomLeftCornerOfText, font, fontScale,fontColor,thickness,lineType)
```

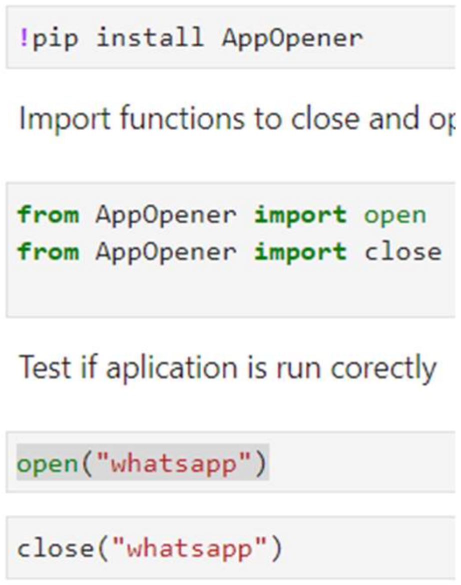




Co-funded by
the European Union

Connecting module with application api

- The module can be used to create precice guests and to execute routines.
- In our case runs whatsup application



```
!pip install AppOpener
```

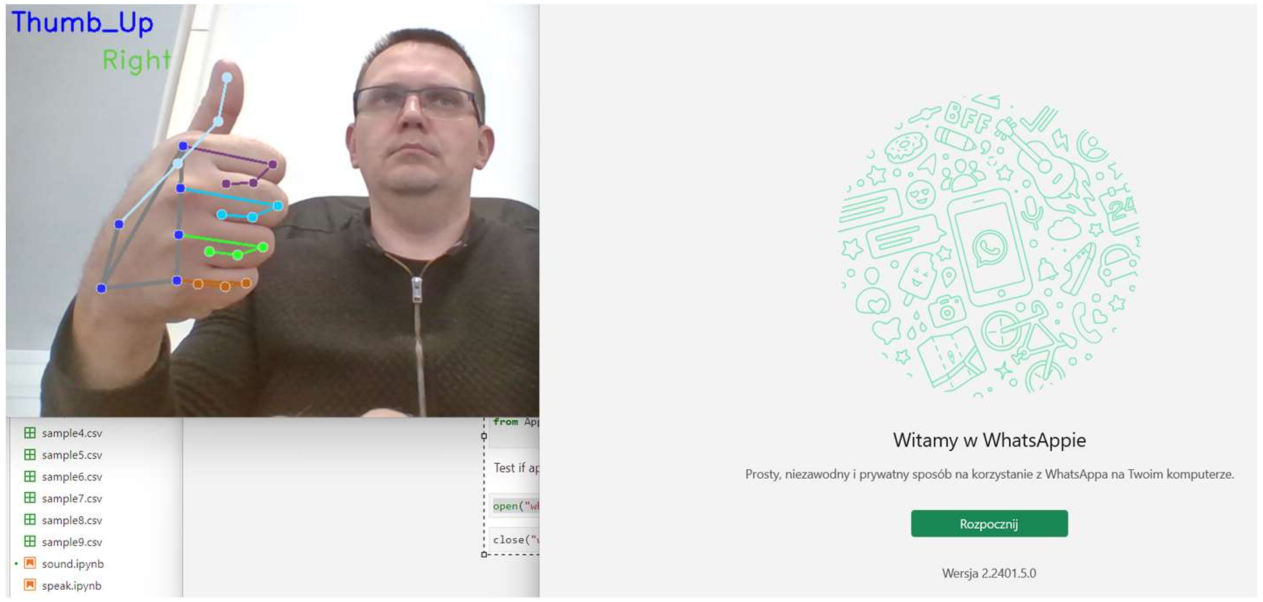
Import functions to close and op

```
from AppOpener import open
from AppOpener import close
```

Test if aplication is run corectly

```
open("whatsapp")
```

```
close("whatsapp")
```



Witamy w WhatsAppie

Prosty, niezawodny i prywatny sposób na korzystanie z WhatsAppa na Twoim komputerze.

Wersja 2.2401.5.0

Co-funded by
the European Union

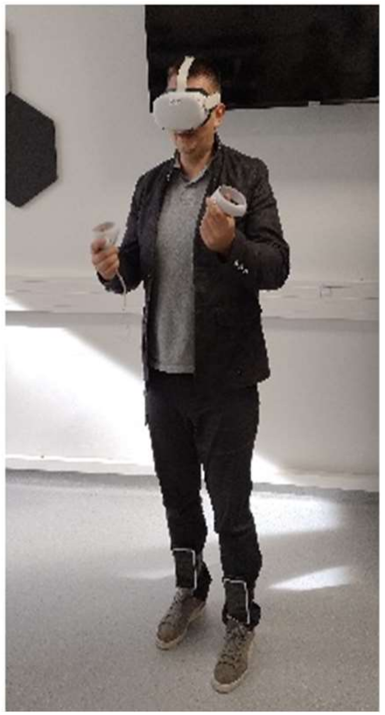
Use case 2 – build own model

- Virtual Reality (VR) is a technology that creates three-dimensional virtual environments for users to explore and interact with objects using special devices like VR goggles and controllers.
- It finds applications in various fields such as military and civilian pilot training, surgical simulation, and education and entertainment.
- VR offers increased immersion by engaging users physically, enhancing their sense of presence in the virtual environment.
- Most VR systems focus primarily on tracking hands and head movements, neglecting lower limb tracking.
- Current systems supporting these features often come with a high price tag.

Co-funded by
the European Union

Leg tracking for VR

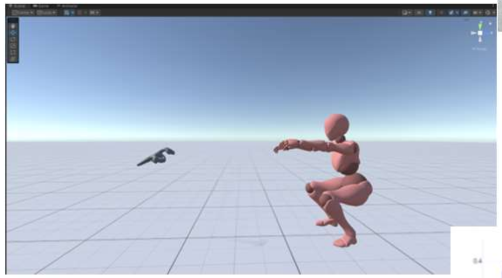
- To enhance game immersion, additional sensors and detectors can be used, but acquiring them entails extra costs.
- This use case proposes utilizing budget smartphones as sensors during gameplay.

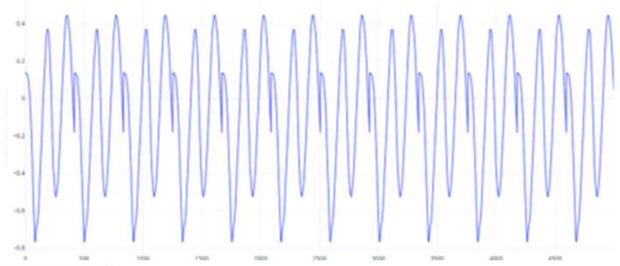


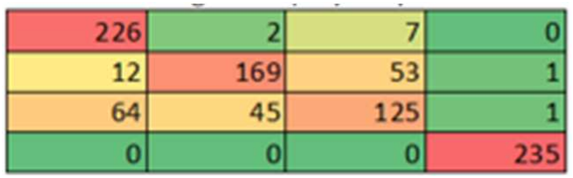
Co-funded by
the European Union

Sensor tracking

- Attach sensors
- Create routine
- Generate results
- Analyse results





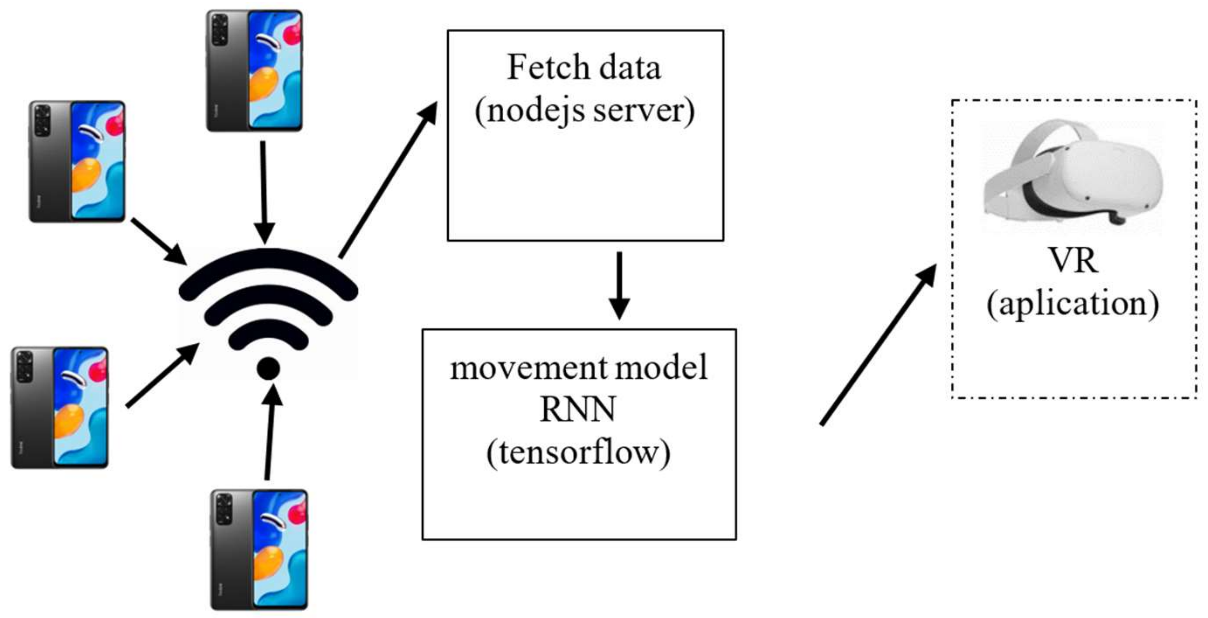


226	2	7	0
12	169	53	1
64	45	125	1
0	0	0	235

Co-funded by
the European Union

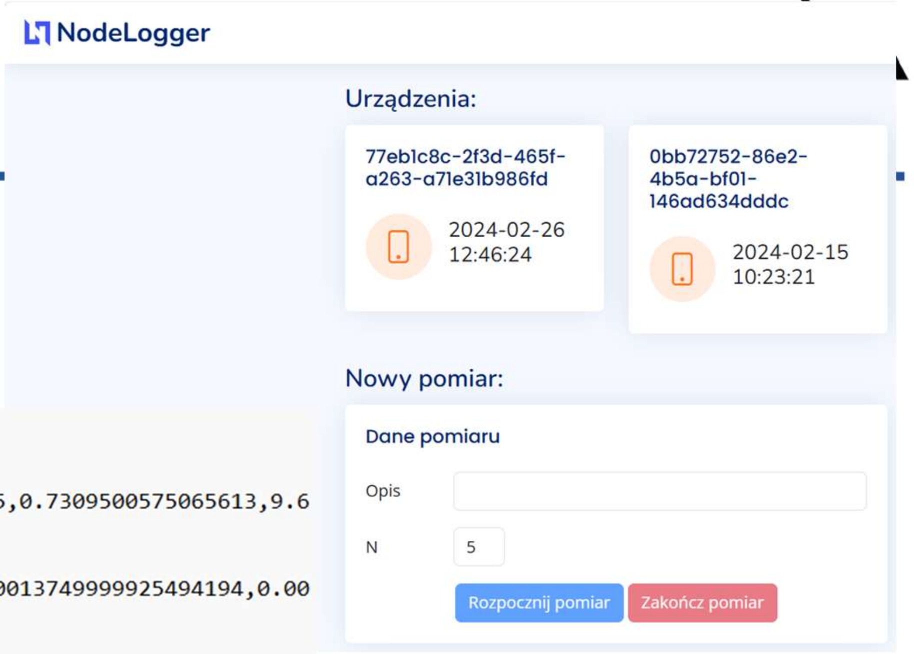
Obtaining data

- The solution is using smartphones as sensors for monitoring limb motion.
- data packets are collected upon application launch and transmitted to a destination for synchronization.



Co-funded by
the European Union

Nodejs app


NodeLogger

Urządzenia:

77eb1c8c-2f3d-465f-a263-a71e31b986fd
2024-02-26 12:46:24

0bb72752-86e2-4b5a-bf01-146ad634dddc
2024-02-15 10:23:21

Nowy pomiar:

Dane pomiaru

Opis

N 5

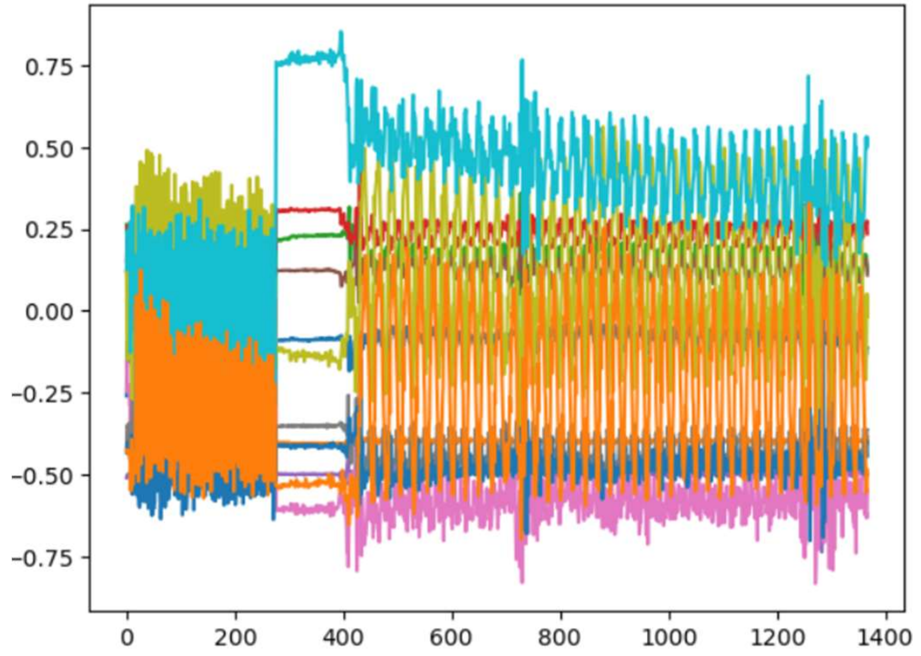
```
"id","deviceId","sensorName","time","x","y","z","marker"
259743,"77eb1c8c-2f3d-465f-a263-
a71e31b986fd","totalacceleration",1708947870967959300,-1.2540000677108765,0.7309500575065613,9.6
55950546264648,"-1"
259744,"77eb1c8c-2f3d-465f-a263-
a71e31b986fd","gyroscope",1708947870967959300,0.0004124999977648258,-0.0001374999925494194,0.00
2337499987334013,"-1"
259745,"77eb1c8c-2f3d-465f-a263-
a71e31b986fd","accelerometer",1708947870967959300,-0.019259000197052956,0.005688999779522419,-0.
0010160000529140234,"-1"
```

```
-0.11620917243219289    -0.4050352775805529     0.24012285934720698     0.28008825639587615     -0.4986388500861196
        -0.09539474190944812    0.7453243954042078      -0.40910578562121863    -0.549179596315432
-0.11594395606935598    -0.4048990083726674     0.23978625736223041     0.280169962616997       -0.4982499656103057
        -0.09835526730880326    0.745555306783839       -0.4103068066392116     -0.5464295735863297
-0.11550192806390687    -0.40496714297661013    0.23953380377733416     0.2815589643054959      -0.4982499656103057
        -0.09720395278616528    0.7453243954042078      -0.4088874260752662     -0.5493629316207216
```

Co-funded by the European Union

Data preprocessing

- Data from smartphone sensors is sampled at frequencies, with 10 Hz being adequate for basic activities and 20 Hz for complex movements. The time-series data is then sent to the inference module.



Co-funded by the European Union

Formating data for TensorFlow

- The data has to be formatted for tensor flow library. The require training data as sequence of samples for each attribute. If model is sequential the each sample has time window.

```
s1 = load_file('sample8.csv')
```

```
s1i = np.diff(s1, axis=0)
```

```
window=8
s1s = subsequences(s1i, window)
```

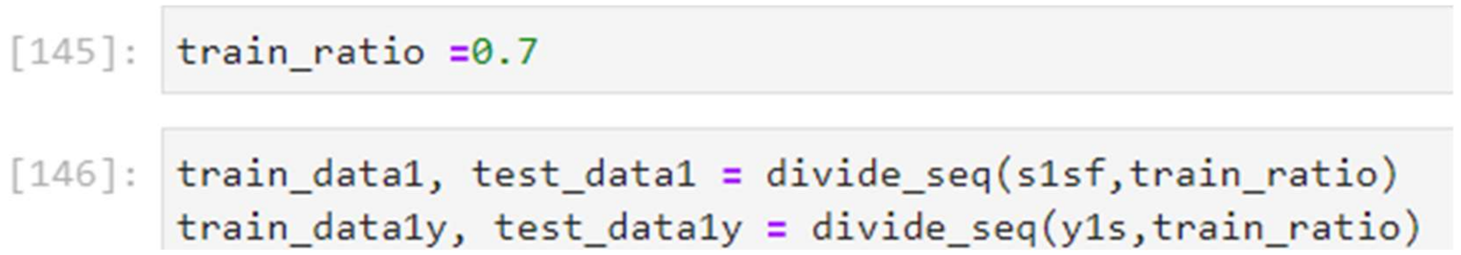
```
s1sf = np.moveaxis(s1s, [0, 1, 2], [1, 2, 0] )
```

```
y1s = np.ones(s1sf.shape[0])*0
```

Co-funded by
the European Union

Training / Verification set

- The data is divided into two subsets for training and verification.
- The training set should be significantly bigger than verification set

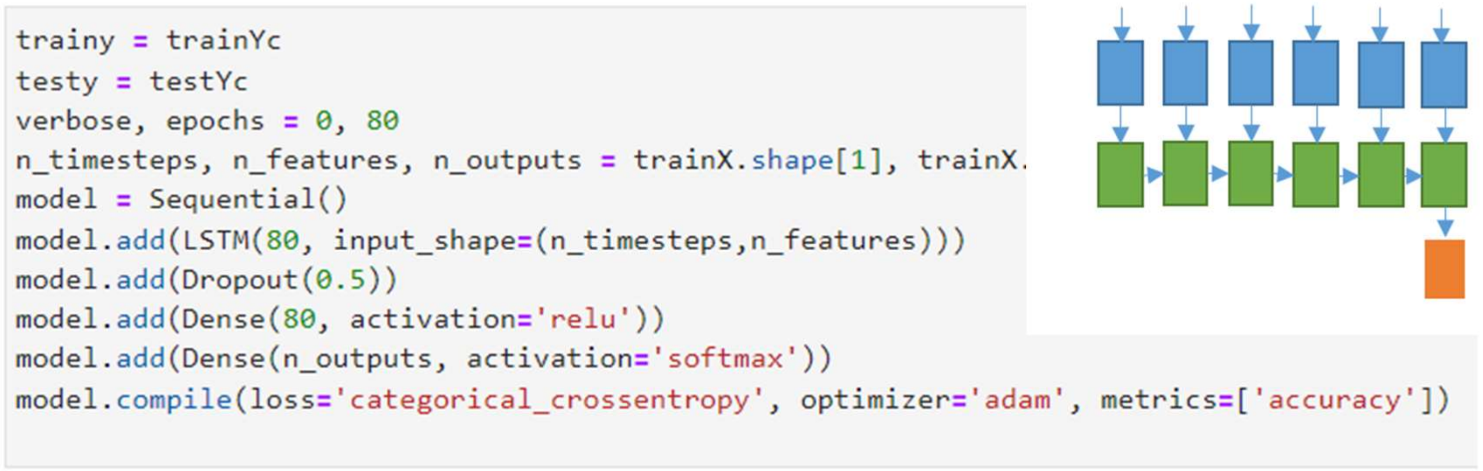


```
[145]:  train_ratio =0.7

[146]:  train_data1, test_data1 = divide_seq(s1sf,train_ratio)
        train_data1y, test_data1y = divide_seq(y1s,train_ratio)
```

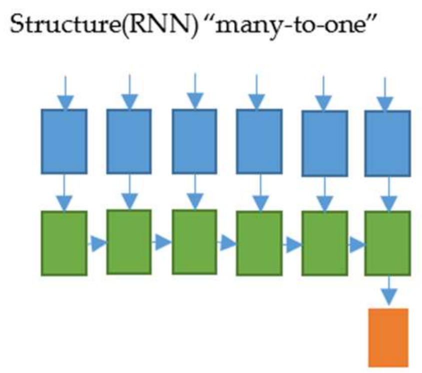
Co-funded by
the European Union

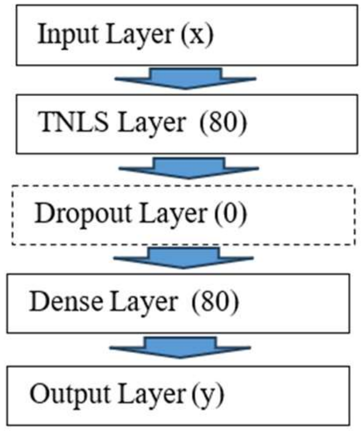
RNN Layer

- The model adjusting for input sequence and class count while reducing neurons for speed, with epochs tuned via learning curve analysis and dropout layers to prevent overfitting.



```
trainy = trainYc
testy = testYc
verbose, epochs = 0, 80
n_timesteps, n_features, n_outputs = trainX.shape[1], trainX.
model = Sequential()
model.add(LSTM(80, input_shape=(n_timesteps,n_features)))
model.add(Dropout(0.5))
model.add(Dense(80, activation='relu'))
model.add(Dense(n_outputs, activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
```

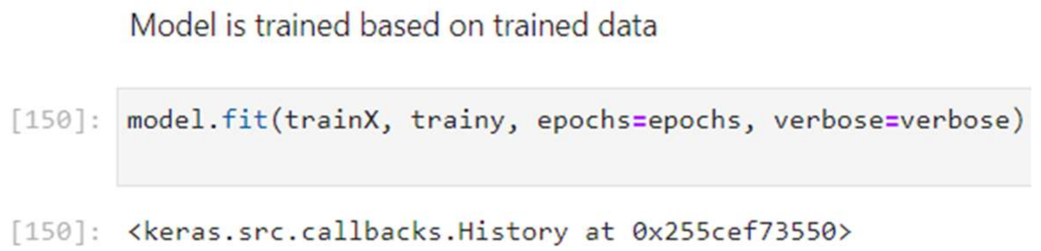




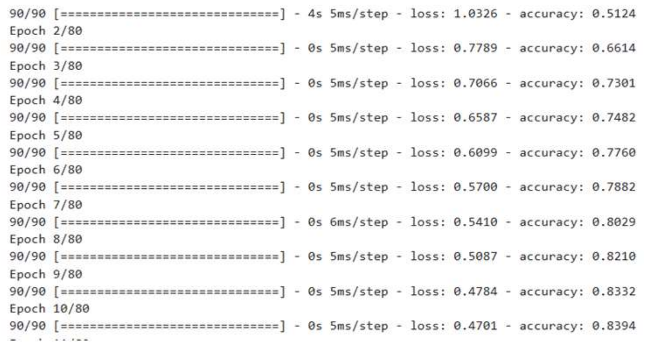
Co-funded by the European Union

Model training and verification

- Training of model is made by feeding it with data

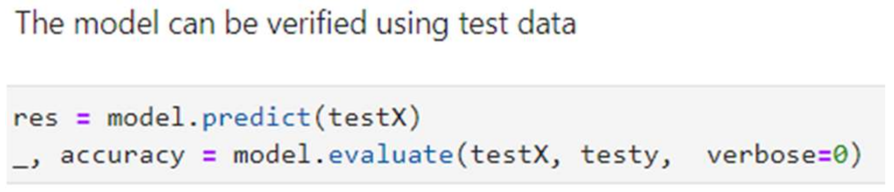

Model is trained based on trained data

```
[150]: model.fit(trainX, trainy, epochs=epochs, verbose=verbose)

[150]: <keras.src.callbacks.History at 0x255cef73550>
```

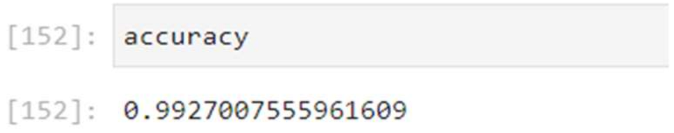


```
90/90 [==============================] - 4s 5ms/step - loss: 1.0326 - accuracy: 0.5124
Epoch 2/80
90/90 [==============================] - 0s 5ms/step - loss: 0.7789 - accuracy: 0.6614
Epoch 3/80
90/90 [==============================] - 0s 5ms/step - loss: 0.7066 - accuracy: 0.7301
Epoch 4/80
90/90 [==============================] - 0s 5ms/step - loss: 0.6587 - accuracy: 0.7482
Epoch 5/80
90/90 [==============================] - 0s 5ms/step - loss: 0.6099 - accuracy: 0.7760
Epoch 6/80
90/90 [==============================] - 0s 5ms/step - loss: 0.5700 - accuracy: 0.7882
Epoch 7/80
90/90 [==============================] - 0s 6ms/step - loss: 0.5410 - accuracy: 0.8029
Epoch 8/80
90/90 [==============================] - 0s 5ms/step - loss: 0.5087 - accuracy: 0.8210
Epoch 9/80
90/90 [==============================] - 0s 5ms/step - loss: 0.4784 - accuracy: 0.8332
Epoch 10/80
90/90 [==============================] - 0s 5ms/step - loss: 0.4701 - accuracy: 0.8394
```

- Verification of model is made be making prediction with test data and checking result


The model can be verified using test data

```
res = model.predict(testX)
_, accuracy = model.evaluate(testX, testy,  verbose=0)
```

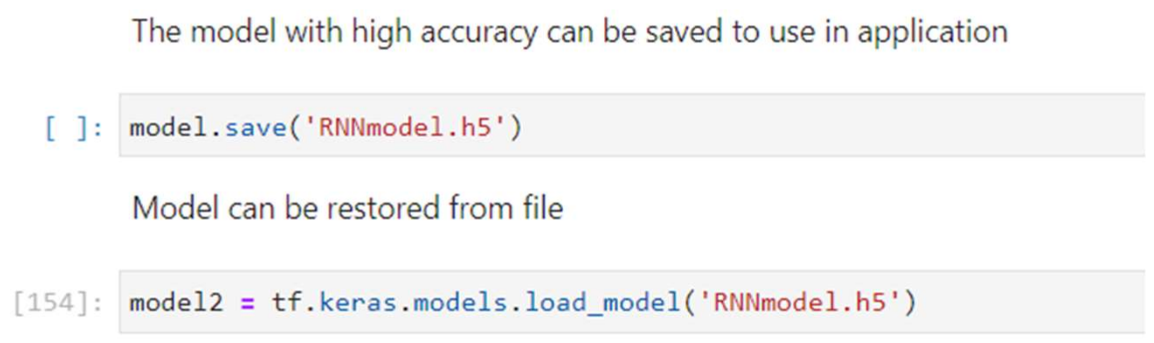


```
[152]: accuracy

[152]: 0.9927007555961609
```

Co-funded by
the European Union

Saving / restoring models

- Created models can be saved / restored to used in other applications
- The tuning process allows to look for better models with all data or part of data



The model with high accuracy can be saved to use in application

```
[ ]: model.save('RNNmodel.h5')
```

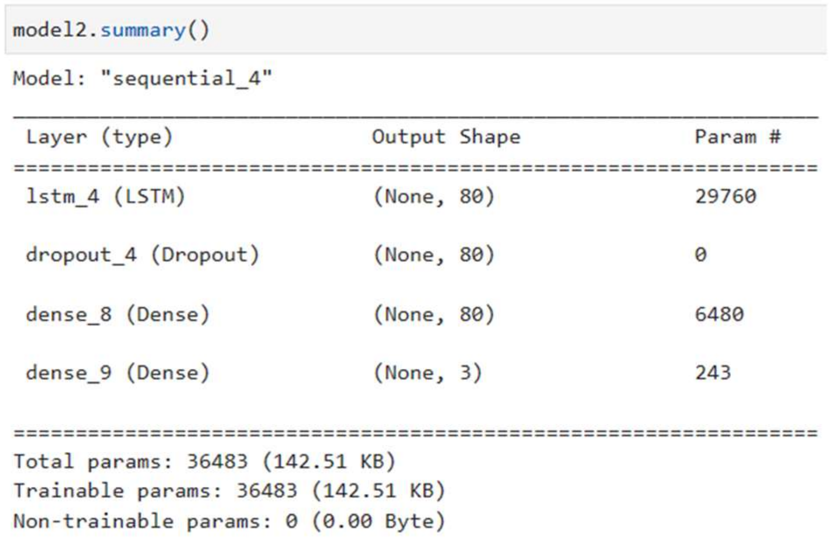
Model can be restored from file

```
[154]: model2 = tf.keras.models.load_model('RNNmodel.h5')
```

Co-funded by
the European Union

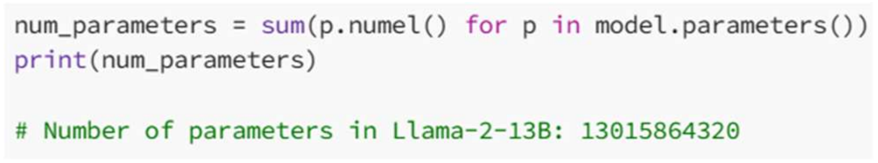
Size of model

- The size of model determine its training time and execution time
- fit time >> prediction time



```
model2.summary()

Model: "sequential_4"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_4 (LSTM)               (None, 80)                29760

 dropout_4 (Dropout)         (None, 80)                0

 dense_8 (Dense)             (None, 80)                6480

 dense_9 (Dense)             (None, 3)                 243

=================================================================
Total params: 36483 (142.51 KB)
Trainable params: 36483 (142.51 KB)
Non-trainable params: 0 (0.00 Byte)
```

```
Yolo3 -image classification
Total params: 62001757 (236.52 MB)
Trainable params: 61949149 (236.32 MB)
Non-trainable params: 52608 (205.50 KB)
```

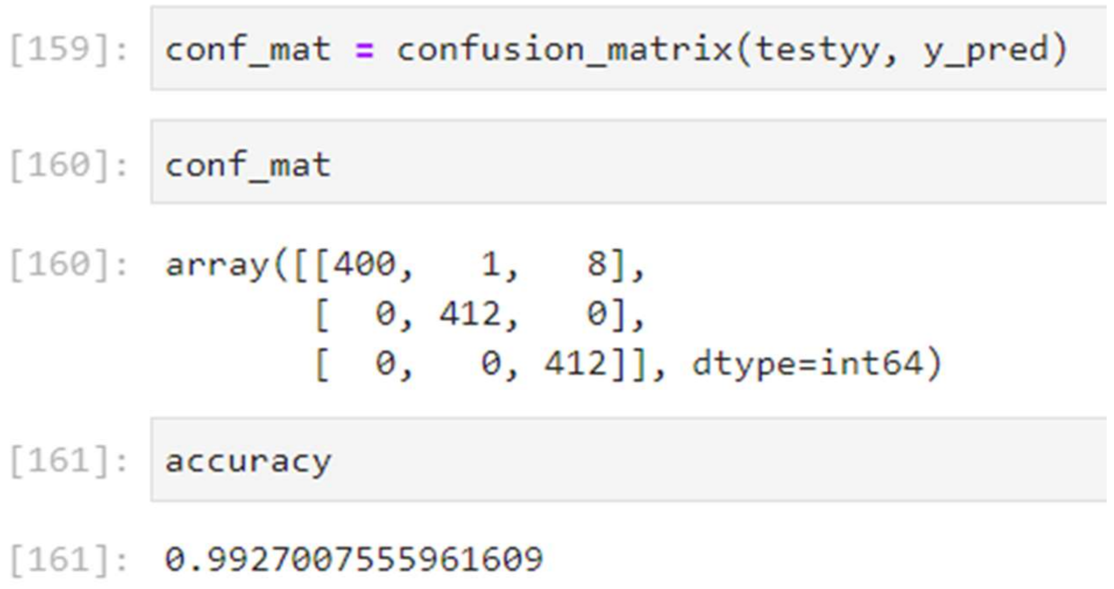


```
num_parameters = sum(p.numel() for p in model.parameters())
print(num_parameters)

# Number of parameters in Llama-2-13B: 13015864320
```

Co-funded by
the European Union

Multiple class classification

- In case of multiple class classification the analysis of each class results can be useful to find class at which a additional data are needed

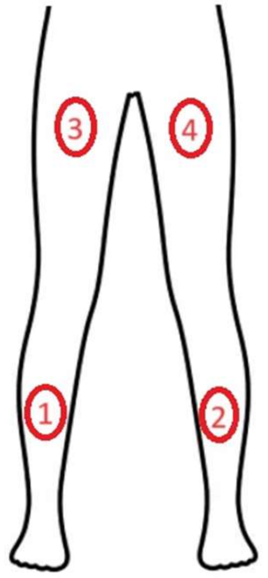


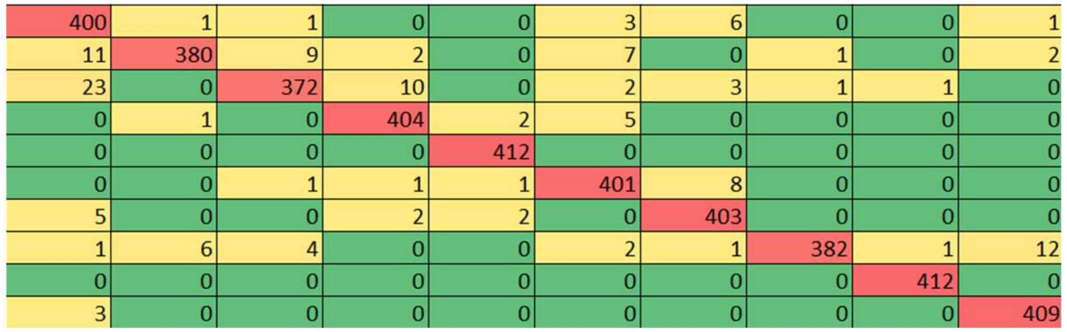
```
[159]: conf_mat = confusion_matrix(testyy, y_pred)

[160]: conf_mat

[160]: array([[400,   1,   8],
              [  0, 412,   0],
              [  0,   0, 412]], dtype=int64)

[161]: accuracy

[161]: 0.9927007555961609
```

Co-funded by
the European Union

Final experiments and results

- solution enables activity tracking with 99% accuracy and player activity detection for ten activity classes within 0.5s.
- detection time to 0.2 seconds by sacrificing detection accuracy to 95% through a window reduction to 5.





400	1	1	0	0	3	6	0	0	1
11	380	9	2	0	7	0	1	0	2
23	0	372	10	0	2	3	1	1	0
0	1	0	404	2	5	0	0	0	0
0	0	0	0	412	0	0	0	0	0
0	0	1	1	1	401	8	0	0	0
5	0	0	2	2	0	403	0	0	0
1	6	4	0	0	2	1	382	1	12
0	0	0	0	0	0	0	0	412	0
3	0	0	0	0	0	0	0	0	409

Co-funded by
the European Union

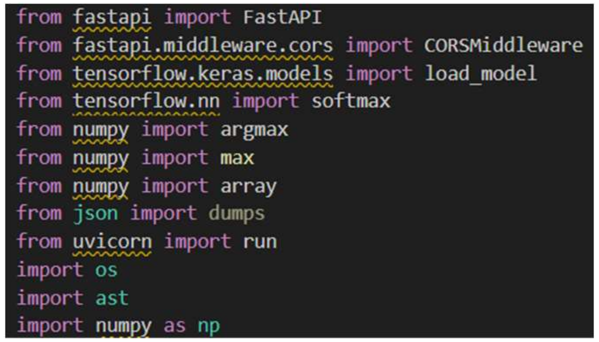
Creating external module

- The AI model can be integrateg with the external module
- The module in case of big model can be hosted by <u>service</u>
- The service interface should be provided e.g. API

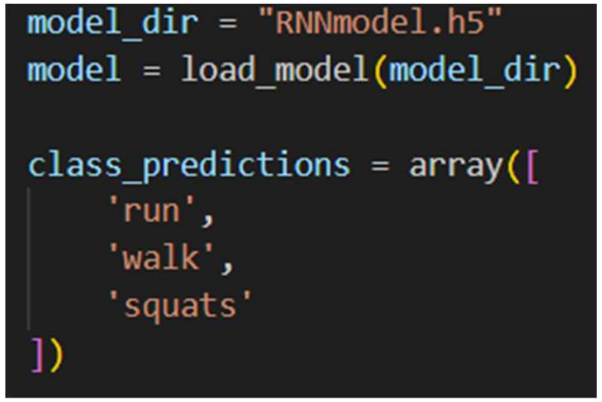
Co-funded by
the European Union

API implementation

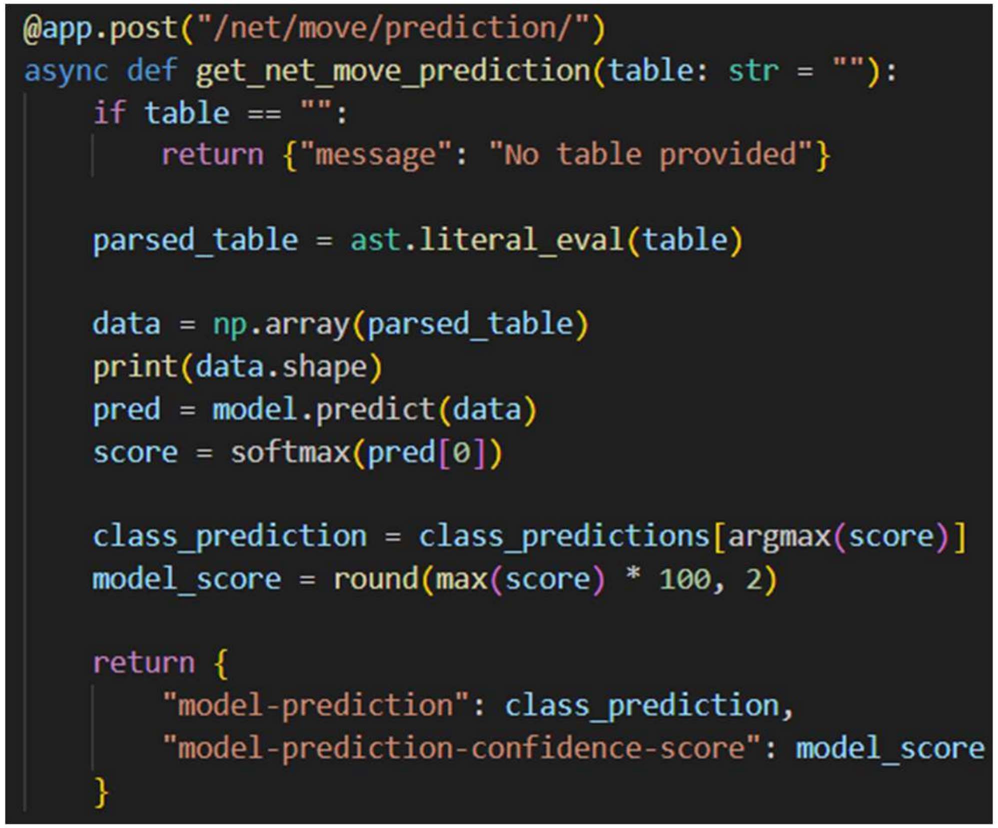
- The fastapi + tensorflow library



```python
from fastapi import FastAPI
from fastapi.middleware.cors import CORSMiddleware
from tensorflow.keras.models import load_model
from tensorflow.nn import softmax
from numpy import argmax
from numpy import max
from numpy import array
from json import dumps
from uvicorn import run
import os
import ast
import numpy as np
```



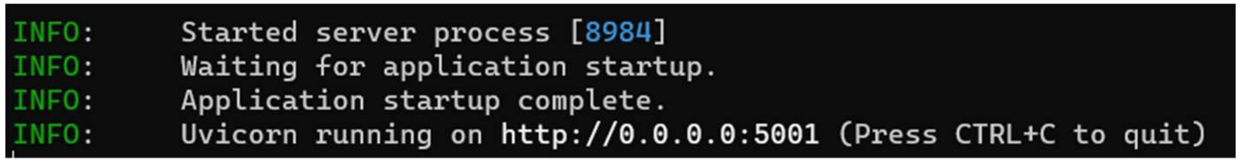
```python
model_dir = "RNNmodel.h5"
model = load_model(model_dir)

class_predictions = array([
    'run',
    'walk',
    'squats'
])
```



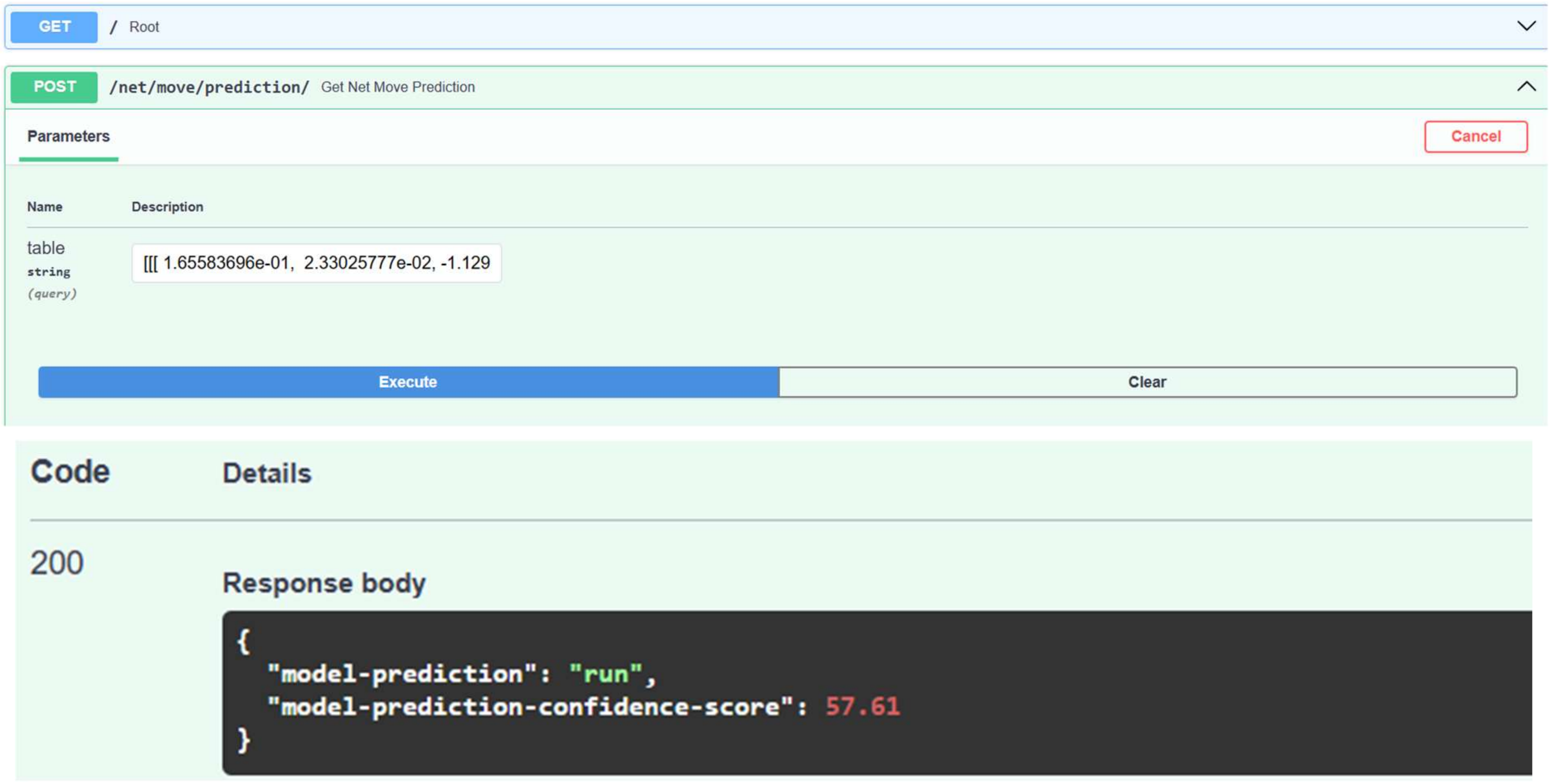
```python
@app.post("/net/move/prediction/")
async def get_net_move_prediction(table: str = ""):
    if table == "":
        return {"message": "No table provided"}

    parsed_table = ast.literal_eval(table)

    data = np.array(parsed_table)
    print(data.shape)
    pred = model.predict(data)
    score = softmax(pred[0])

    class_prediction = class_predictions[argmax(score)]
    model_score = round(max(score) * 100, 2)

    return {
        "model-prediction": class_prediction,
        "model-prediction-confidence-score": model_score
    }
```

Co-funded by
the European Union

API example



```
INFO:     Started server process [8984]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
INFO:     Uvicorn running on http://0.0.0.0:5001 (Press CTRL+C to quit)
```



GET / Root

POST /net/move/prediction/ Get Net Move Prediction

Parameters

Name	Description
table string (query)	[[[1.65583696e-01, 2.33025777e-02, -1.129

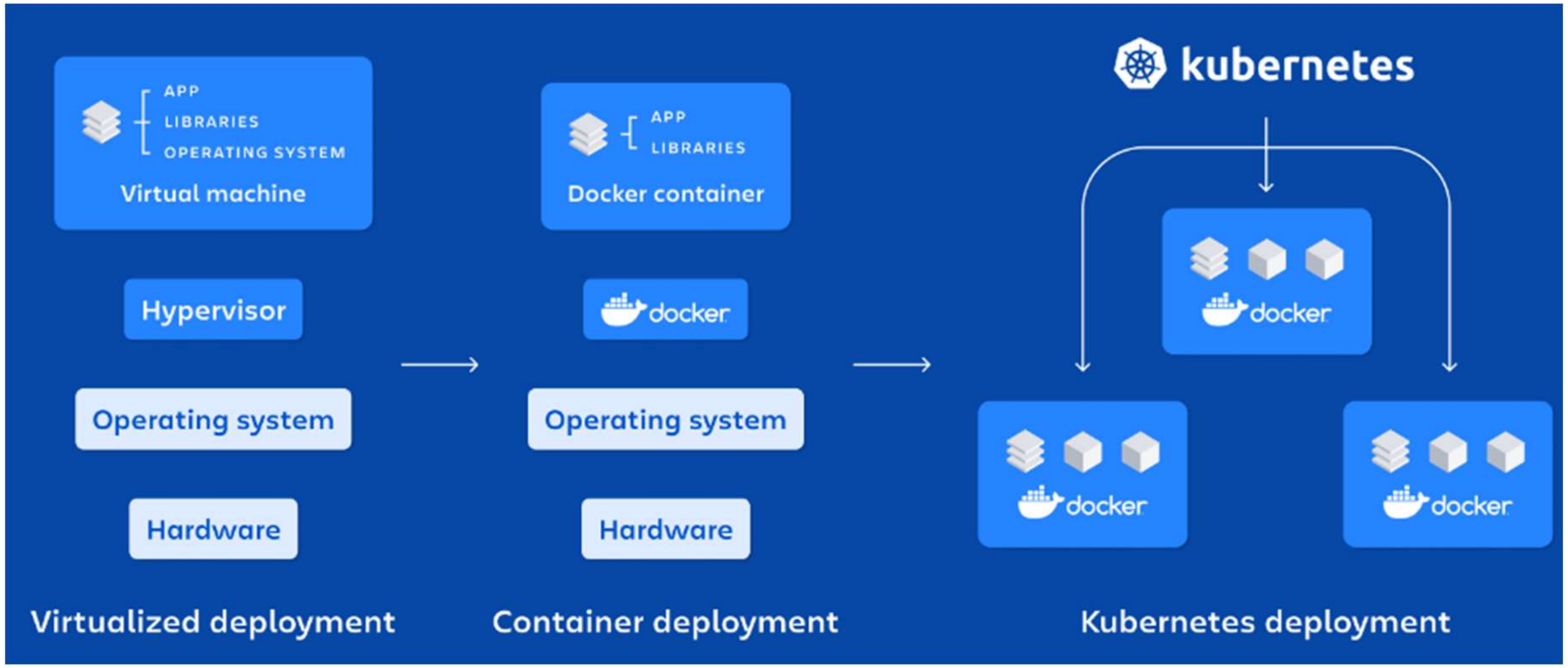
Code	Details
200	Response body

```
{
  "model-prediction": "run",
  "model-prediction-confidence-score": 57.61
}
```

Co-funded by the European Union

Next step

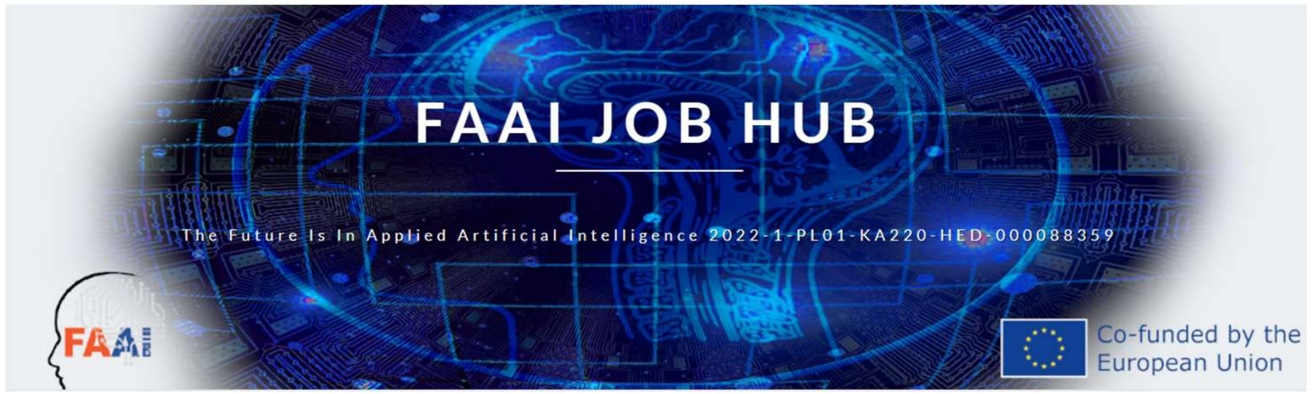
- Creating image using docker
- Load balancing using Kubernetes



[https://www.atlassian.com/microservices/microservices-architecture/kubernetes-vs-docker]

Co-funded by
the European Union

FAAI project


FAAI JOB HUB
The Future Is In Applied Artificial Intelligence 2022-1-PL01-KA220-HED-000088359
FAAI
Co-funded by the European Union

ABOUT US

Welcome to **FAAI Smart JobHub**

within **Erasmus+** project: **FAAI JOB HUB The Future Is In Applied Artificial Intelligence**

Project number: **2022-1-PL01-KA220-HED-000088359**

https://faai.ath.edu.pl/

Co-funded by the European Union