
Virtual Walk

Release 0.1

Oct 02, 2020

Contents:

| | | |
|----------|----------------------------|-----------|
| 1 | Data Processing | 3 |
| 2 | Person entity | 5 |
| 3 | Feed Forward Model | 7 |
| 4 | Firefox Controller | 9 |
| 5 | Webcam Wrapper | 11 |
| 6 | Indices and tables | 13 |
| | Python Module Index | 15 |
| | Index | 17 |

Hola me llamo Gonzalo.

CHAPTER 1

Data Processing

CHAPTER 2

Person entity

```
class source.entities.person.Person (heatmap=None, offsets=None, rescale=(1, 1), thresh-  
                                     old=0.7, path_txt=None, show_head=False, out-  
                                     put_stride=32)
```

Person detected in a frame. With this class a person entity can be extracted from an image. All the keypoints are extracted and keypoints saved.

Returns: Person:

get_height ()

keypoints: 15: LEFT FOOT, 16: RIGHT FOOT, 0: NOSE. :return:

get_width ()

keypoints: 5: LEFT SHOULDER, 6: RIGHT SHOULDER. :return:

infer_lc_keypoints (*prev_person*)

This function will be used when creating the frame groups. It takes the Person of the previous frame and uses its keypoints to infer the keypoints of “self”. If the keypoint has low confidence, the position of the keypoint relative to the neck is used in the current person

Args: prev_person (Person): The person from the current frame

infer_point (*index, prev_person*)

Use the position of the neck and the same keypoint from the previous frame to infer this one. Same confidence is applied

Args: index (int): Index of the keypoint to infer prev_person (Person): Person from the previous frame

is_valid_first ()

This function determines if the frame should be considered for training. Before it was embedded inside the pipeline of DataProcessor. Now it's a function, so conditions can be changed based on performance.

Returns: bool: True if is valid

```
class source.entities.person_frames.PersonMovement (list_persons,          times_v=10,  
                                                    joints_remove=(13, 14, 15,  
                                                    16), model='LSTM')
```

Extracts coordinates from a list of persons. The coordinates extracted are the input for all the models that predict actions.

It has two main functions:

- `get_vector`: Get positions of keypoints and joint speeds. Used for Non LSTM models
- `get_vector`: Get positions of keypoints. Input for LSTM model.

Returns: PersonMovement:

`get_vector` (*times_v, joints_remove*)

Get coordinates vector from a series of frames.

Args: `times_v` (int): Times the body velocity is repeated in the resulting vector. `joints_remove` (tuple): Joints that will be removed and not used in the final vector

Returns: ndarray: Flattened vector of $[x + v * \text{times_v} + v]$ dimensions. Where `x` and `v` are the flattened vectors of joints positions and velocities.

`get_vector_lstm` (*joints_remove*)

Get coordinates vector (only positions) from a series of frames.

Args: `joints_remove` (tuple): Joints that will be removed and not used in the final vector

Returns: ndarray: Flattened vector of $[x]$ dimensions. Where `x` is the flattened vector of joints positions and velocities.

CHAPTER 3

Feed Forward Model

CHAPTER 4

Firefox Controller

```
class source.controller.Controller (classes,          initial_url=None,      driver_path=None,  
                                  time_rotation=0.5, coordinates=None)
```

Class to control the movement in Google Street View. It only works with Firefox. If there is no guess about where to start (*initial_url* or *coordinates*) it starts in Zaragoza (Spain).

Returns: Controller:

```
perform_action (i_action)
```

Perform an action given its index within *self.classes*

Args: *i_action* (int): index of classes list

```
perform_action_name (action)
```

Perform action given its name.

Args: *action* (str): Name of the action to perform. As it uses a mapper, it must be one of ['walk', 'stand', 'left', 'right']

```
class source.controller.DistanceMeter (starting_url=None,          starting_coords=None,  
                                       units='km')
```

This class is used to keep track of the distance made during a walk. It retrieves coordinates from a google maps url using regex and calculates distance walked.

```
static distance_from_coords (coords1, coords2)
```

Gets two pairs of coordinates in decimal format and calculates distance between points

Args: *coords1* (list): Coordinates in decimal format (lat, long) *coords2* (list): Coordinates in decimal format (lat, long)

Returns: float: Meters between points

```
retrieve_coords (url)
```

From a Google Maps URL it obtains the coordinates

Args: *url* (str): URL from which coordinates have to be extracted

Returns: list: Coordinates in decimal format (lat, long)

update_distance (*url*)

self.distance keeps record of all the distance made in the walk. This function updates this value.

Args: url (str): URL of the new step

CHAPTER 5

Webcam Wrapper

CHAPTER 6

Indices and tables

- `genindex`
- `modindex`
- `search`

S

`source.controller`, 9
`source.entities.person`, 5
`source.entities.person_frames`, 5

C
Controller (class in source.controller), 9

D
distance_from_coords() (source.controller.DistanceMeter static method), 9
DistanceMeter (class in source.controller), 9

G
get_height() (source.entities.person.Person method), 5
get_vector() (source.entities.person_frames.PersonMovement method), 6
get_vector_lstm() (source.entities.person_frames.PersonMovement method), 6
get_width() (source.entities.person.Person method), 5

I
infer_lc_keypoints() (source.entities.person.Person method), 5
infer_point() (source.entities.person.Person method), 5
is_valid_first() (source.entities.person.Person method), 5

P
perform_action() (source.controller.Controller method), 9
perform_action_name() (source.controller.Controller method), 9
Person (class in source.entities.person), 5
PersonMovement (class in source.entities.person_frames), 5

R
retrieve_coords() (source.controller.DistanceMeter method), 9

S
source.controller (module), 9
source.entities.person (module), 5
source.entities.person_frames (module), 5

U
update_distance() (source.controller.DistanceMeter method), 9