# Manual to Berlin-APC: A Privacy-Friendly Dataset for Automated Passenger Counting in Public Transport

Robert Seidel, Dimitra Zarafeta, Michael Siebert, Ramtin Dastgheib Shirazi,
Sambu Seo, Thomas Goerttler,† Klaus Obermayer

## I. Introduction

This document provides a short technical introduction to the *Berlin-APC* dataset (version 2). The dataset consists of two files, a HDF5 file which contains the image sequences, and a CSV file which contains the labels. The CSV file has three columns: (1) the image sequence name; (2) the number of boarding passengers in that image sequence; (3) the number of alighting passengers in that image sequence. The image sequence names also serve as keys in the HDF5 file. The HDF5 file's datasets (indexed by the aforementioned sequence names) are `float16` arrays of the shape `(number of frames, 20, 25)`, the pixel values range between 0–1, and the frame rate is 10 frames per second.

```
$ head berlin-apc.csv
000000_000_01_06,3,0
000001_000_01_07,0,2
000002_000_01_08,0,1
000003_000_01_09,1,0
000004_000_01_09,1,3
000005_000_01_09,1,0
000006_000_01_10,0,1
```

The **image sequence name** is structured as follows:

`<serial number>_<recording day id>_<recording month>_<recording hour>`

For example, the entry `012948_030_09_15,0,3` indicates the $12,948^{\text{th}}$ image sequence of the dataset (counting starts at zero),[1] which was recorded on a day in September 2017, between 3.00 and 4.00 pm. That day is the $30^{\text{th}}$ recording day, i.e., all image sequence with a recording day id of 30 were recorded on the same day. According to manual counting, this image sequence shows three alighting passengers, and no boarding passengers.

## II. Working with the HDF5 and CSV File

We now give an example how to work with the dataset in Python:

```
import pandas as pd
import numpy as np
import h5py as h5
import matplotlib.pyplot as plt

# Load HDF5 and CSV file.
sequences = h5.File("berlin-apc.h5", mode="r")
labels = pd.read_csv("berlin-apc.csv", names=["sequence_name", "n_boarding", "n_alighting"])

# Inspect CSV file: Print first 5 labels.
print(labels.head())
>>>       sequence_name  n_boarding  n_alighting
>>> 0  000000_000_01_06           3            0
>>> 1  000001_000_01_07           0            2
>>> 2  000002_000_01_08           0            1
>>> 3  000003_000_01_09           1            0
>>> 4  000004_000_01_09           1            3
```

R. Seidel, D. Zarafeta and R.D. Shirazi were funded by a grant from the European Regional Development Fund to K. Obermayer (grant number 10167006). This work was also funded by a European Regional Development Fund grant to Interautomation Deutschland GmbH (grant number 10166961).

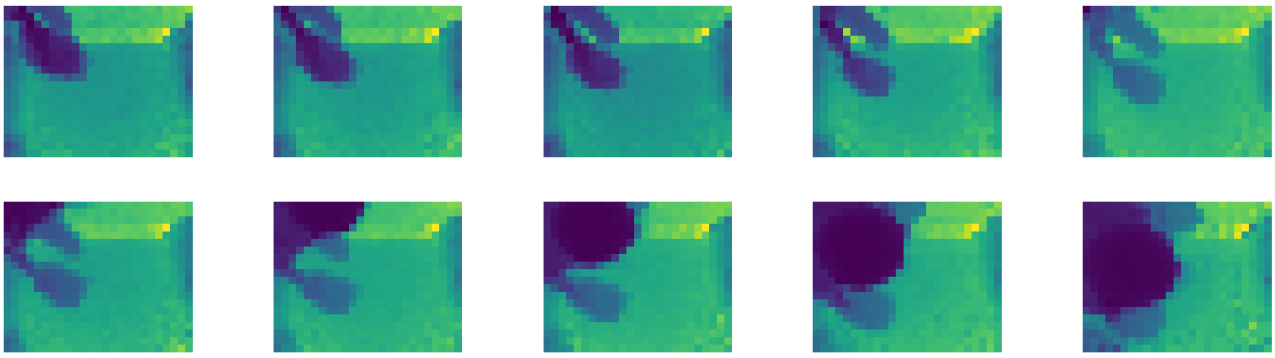
Michael Siebert is with Interautomation Deutschland GmbH, Hauptstraße 56–60, 13158 Berlin, Germany. The other authors are with Technische Universität Berlin, Institute of Software Engineering and Theoretical Computer Science, Neural Information Processing Group, Straße des 17. Juni 135, 10623 Berlin, Germany. Dagger symbol indicates corresponding author (thomas.goerttler@tu-berlin.de).

The authors wish to thank the many labeling assistants for their help in viewing and annotating the video material.

[1]Note that the serial numbers are not consistent between different versions of this dataset.

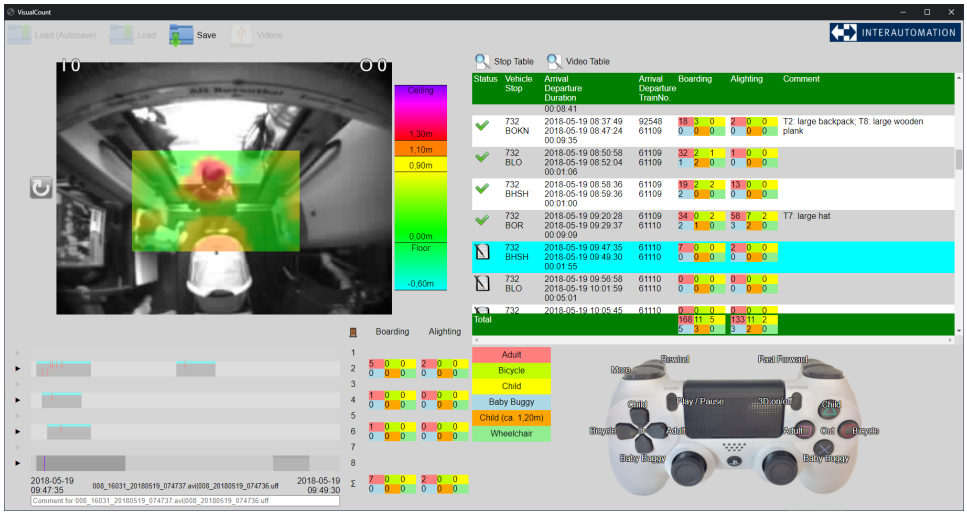
```
print(len(labels))
>>> 12956

# Inspect HDF5 file: Load the first image sequence.
sequence = sequences["000000_000_01_06"]
print(sequence.__class__)
>>> <class 'h5py._hl.dataset.Dataset'>

print(sequence.shape)
>>> (243, 20, 25)

# Get the 90-99th frame and save it as png file.
for i in range(90, 100):
    frame = np.array(sequence[i, :, :], dtype="float32")
    plt.imshow(frame)
    plt.axis("off")
    plt.savefig(f"frame{i}.png")
```

The frames 90–99 from the previous example look like this:



## III. Annotation Details and Software

The video annotation software *VisualCount* (developed by INTERAUTOMATION Deutschland GmbH, Berlin, Germany; see screenshot below) was used to perform the manual counts. The labels for this dataset have been created using both low resolution 3D videos (contained in the HDF5 file) as well as high resolution 2D videos (not included in this dataset) to make the scenes more comprehensible for human annotators. *VisualCount* renders the 3D information as overlay into the 2D footage (see the left part of the screenshot below, as well as the color scheme right of the 2D/3D overlay). The height information from 3D videos was used to distinguish between children and adults with a threshold height of 1.20 meters. In the screenshot, the small person in the center of the frame would be considered an adult, while the person in the left a child. A game controller is used to navigate through the video with playback or rewind speeds corresponding to the pressure applied to the buttons to obtain both slow motion for crowded scenes as well as high speeds (10x and more) to skip through long idle timespans (see bottom right of the screenshot for the assignment of keys).



## IV. How to cite

Please use the following template to cite this paper:

```
@online{berlinapc,
  author =  {Seidel, R.
             and Zarafeta, D.
             and Siebert, M.
             and Dastgheib Shirazi, R.
             and Seo, S.
             and Goerttler, T.
             and Obermayer, K.},
  title = {Manual to Berlin-APC:
           A Privacy-Friendly Dataset for Automated Passenger Counting in Public Transport},
  url = {http://dx.doi.org/10.14279/depositonce-12205.3},
  organization = {Technische Universit\"at Berlin, Interautomation Deutschland GmbH},
  year = {2021}
}
```

## V. License



EUROPÄISCHE UNION
Europäischer Fonds für regionale Entwicklung

This work was funded by the European Regional Development Fund.