CS230

# Person Re-ID for Follow-Me Task

**Sarah Brennan**
Department of Mechanical Engineering
Stanford University
sbrenn@stanford.edu

## Abstract

The task of correctly re-identifying people after periods of absence is a challenging computer vision problem. This project focuses on applying CNNs to learn feature representations of people and then use nearest neighbor clustering for robust identification of individuals.

## 1 Introduction

This project focuses on the task of re-identifying a target after it is lost from view. The task is intended to be implemented on the JackRabbot (JR), the social navigation robot. The goal is to acquire a target and then follow the target around different spaces at a set distance. This project addresses the specific challenge of re-identifying the target after they are lost from view. The difficulty of this task comes from maintaining a consistent target ID even if the target turns a corner, another person crosses the robot's path, or the target is occluded. Successful completion of this task would be useful for navigating crowded spaces with JR.

## 2 Related work

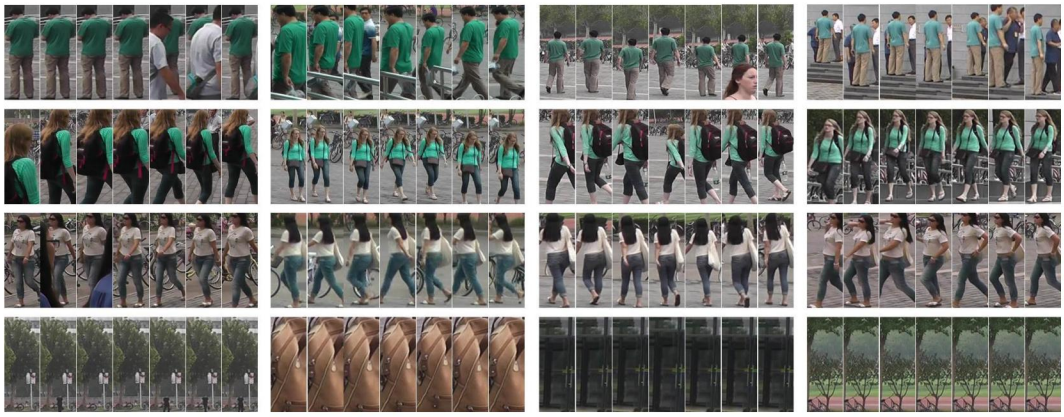
This task is related to traditional person re-ID which involves several cameras trying to identify people from different positions. Traditional re-ID requires different cameras to recognize certain features of individuals that are consistent across different distances, lighting, camera angles, and time stamps. Extensive work has been done in this area for identifying individuals for surveillance purposes with CNNS learning from data sets such as the Motion Analysis and Re-identification Set (MARS) [1]. Another component of the re-ID problem is identifying targets in real time, which typically requires a k-nearest neighbors type approach to group objects with similar features for classification.

Current approaches include Deep SORT[2] and Faster R-CNN. These networks learn a set of feature descriptors fed based on bounding box data from another model such as YOLO. So a two step approach is used. The Follow-Me version of Re-ID is slightly different in that once individuals are detected a target must be determined and then followed as the environment changes. This task requires an even more robust person ID, and this model must be learned online, since the target is not always the same individual. Therefore, this task adds on one more layer to the machine learning pipeline. See the next Section 4 for more detail on the architecture.

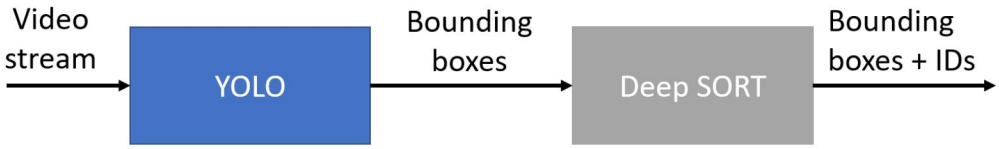
## 3 Dataset and Features

For my model I used the existing MARS data set as a baseline to test my model. This included video data with corresponding bounding box detections for each frame. Video data was collected across six cameras and included tracks for individuals across time. See below for an example [1].

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)



I tried generating my own data set using YOLOv3 to generate bounding boxes, however the output format of the bounding boxes did not match the input format necessary for the Deep SORT model. Additionally, I tried using an existing framework for a YOLO to Deep SORT pipeline (see below for outline), however I was unable to import my trained models. It was not implementable without modification, so I decided to focus on training the model and comparing the output results, rather than creating a custom pipeline. I noticed that this task in general required many custom models stitched together with custom features. It seemed like having more of an end to end model technique could help, however this was outside of the scope of the project due to time constraints.




## 4 Methods

The Deep SORT architecture consisted of a CNN to learn feature representations and nearest neighbor classifier to assign IDs to the bounding boxes. The CNN used a wide residual network to learn 128d feature vectors offline based off large people re-ID data sets such as MARS. The feature vectors were fed to the classifier. A Kalman filter was used to take into account the short term motion of the bounding box (d1 in the equation below) and the cosine similarity of the feature vectors (d2 in the equation below). This similarity metric helped the model re-identify individuals after occlusion. The similarity metric also stored a gallery of previous feature vectors associated with a particular track to help with re-identification as well.

$$c_{i,j} = \lambda d^{(1)}(i,j) + (1-\lambda) d^{(2)}(i,j)$$

The original CNN architecture consisted of two convolution layers, followed by a max pooling layer, then 6 residual blocks, a dense layer, and a final normalization to create the 128d feature vector [3]. This network had over 2.8 million parameters and was trained prior to implementation. The wide residual blocks allowed for faster training. Dropout was used to ensure information was learned during training. The network focuses on learning mid-level features which is the current trend in person re-ID. Below is an outline of the CNN structure [3].

| Layer | Filter Size | Stride | Output |
|---|---|---|---|
| Conv 1 | 3 x 3 | 1 | 32 x 128 x 64 |
| Conv 2 | 3 x 3 | 1 | 32 x 128 x 64 |
| Max Pool 3 | 3 x 3 | 2 | 32 x 64 x 32 |
| Residual 4 | 3 x 3 | 1 | 32 x 64 x 32 |
| Residual 5 | 3 x 3 | 1 | 32 x 64 x 32 |
| Residual 6 | 3 x 3 | 2 | 64 x 32 x 16 |
| Residual 7 | 3 x 3 | 1 | 64 x 32 x 16 |
| Residual 8 | 3 x 3 | 2 | 128 x 16 x 8 |
| Residual 9 | 3 x 3 | 1 | 128 x 16 x 8 |
| Dense 10 | | | 128 |
| Batch and $l_2$ norm | | | 128 |

## 5 Experiments/Results/Discussion

Both the classifier and CNN were modified to address re-ID after occlusions. The gallery was expanded on the classifier to store more feature vectors associated with a particular track. The CNN model was modified to have an additional convolutional layer to increase complexity, and another version added a shortcut after the max pooling layer all the way to output for learning features at different layers similar to OSNet [4].

The gallery modification was tested on the MOT16 data set. One of the important metrics used in the MOT16 evaluation that was relevant for the follow-me task was the ID switching metric. This metric kept track of the number of ID switches that occurred during a video segment, which aligned with the goal of making the model more robust to occlusions and ID switching.

The CNN model was trained using the MARS data set and was to be evaluated against the MARS metrics, however this method of evaluation only tested the accuracy of the feature representation, and not the whole system including the assignment of the IDs with the NN classifier. In the future, the model should be integrated into the Deep SORT system and evaluated against the MOT metrics as well.

## 6 Conclusion/Future Work

Future work includes evaluating the new CNN models against the MOT16 metrics. Also trying transfer learning using data collected from JR, and incorporating target identification. Additionally, attempting some of the unsupervised learning models, and trying an end-to-end approach rather than stacking many custom features. One of the main challenges with the particular framework is integrating all the different models from object detection through ID, at each step custom modules are needed to connect the models.

## 7 Contributions

Started with working on implementing a YOLO to Deep SORT framework to test on my own machine. Attempted this first so that I could use YOLO to create bounding boxes on my own data and test the models success directly with Deep SORT. However, the output format of YOLO's bounding boxes does not match Deep SORT's input format, so a custom module is needed to connect the two. The existing repos for YOLO to Deep SORT did not run in their current state and needed further modification to work. So I decided to test with the already existing datasets (MARS and MOT). I trained 3 cosine metric CNN models on the MARS dataset, however the evaluation software was separate and due to machine incompatibilities I could not run. I ran into similar issues when trying to run evaluations using the MOT challenge software.

## References


[1] @proceedingszheng2016mars, title=MARS: A Video Benchmark for Large-Scale Person Re-identification, author=Zheng, Liang and Bie, Zhi and Sun, Yifan and Wang, Jingdong and Su, Chi and Wang, Shengjin and Tian, Qi, booktitle=European Conference on Computer Vision, year=2016, organization=Springer

[2] @inproceedingsWojke2017simple, title=Simple Online and Realtime Tracking with a Deep Association Metric, author=Wojke, Nicolai and Bewley, Alex and Paulus, Dietrich, booktitle=2017 IEEE International Conference on Image Processing (ICIP), year=2017, pages=3645–3649, organization=IEEE, doi=10.1109/ICIP.2017.8296962

[3] @inproceedingsWojke2018deep, title=Deep Cosine Metric Learning for Person Re-identification, author=Wojke, Nicolai and Bewley, Alex, booktitle=2018 IEEE Winter Conference on Applications of Computer Vision (WACV), year=2018, pages=748–756, organization=IEEE, doi=10.1109/WACV.2018.00087

[4] @articlezhou2019osnet, title=Omni-Scale Feature Learning for Person Re-Identification, author=Zhou, Kaiyang and Yang, Yongxin and Cavallaro, Andrea and Xiang, Tao, journal=arXiv preprint arXiv:1905.00953, year=2019