# Person Re-Identification in a Video Sequence

Zhiyuan Li
Stanford University
zhiyuanl@stanford.edu

Jiayang Wang
Stanford University
jw4149@stanford.edu

## Abstract

*In this paper, we study the task of person re-identification in a video sequence with deep learning methods. We trained and refined a novel CNN based model, the OSNet, with two popular datasets with transfer learning techniques and performed cross-evaluations on the datasets. We then built a practical application that takes any video sequence and a query as input, and outputs all occurrences of the target query in the video sequence. Using OSNet as a feature extractor, we analyze and compare the performance of the original and refined models on the application with real-world examples.*

## 1. Introduction

Person re-identification is a computer vision task that aims to detect and identify a person of interest with various poses and orientations across different locations. This task is crucial for applications such as surveillance, smart cities, and autonomous vehicles. Persons were usually identified through their dressing and appearances [8] [2]. Key challenges for this task include viewpoint variations, pose variations, lighting conditions, occlusions, and other people with similar appearances [16]. Methods for this application usually leverage feature representations that does a good summarizing of the target [5] [3]. In recent years, deep learning based algorithms were developed to learn feature representations that are robust to challenging variations and changing environments [1]. In this paper, we study a novel CNN based Re-ID model, the OSNet [21] [22] that is capable of extracting rich feature representations. We train the model on the Market1501 dataset [16], and refine the learnt model on the DukeMTMC-reID dataset [12] through transfer learning techniques. We then apply the model to build a generalized and robust application that is capable of identifying and tracking persons of interest in a video sequence, such as a security footage. In the process, we also perform analysis on how transfer learning has empowered the refined model to perform better through real-world examples. We trained our models and built the application on top of Torchreid [20], the code base for the original research on OSNet [21] [22] https://github.com/KaiyangZhou/deep-person-reid.

### 1.1. Models

OSNet, a novel CNN-based Re-ID model, captures omni-scale feature representations using Aggregation Gate mechanisms and residual blocks. This design addresses challenges like viewpoint and pose variations. The use of lightweight Depthwise Separable Convolutions ensures efficient computation, making OSNet a fit for re-ID surveillance and smart city applications.

Initially, we trained OSNet on Market1501, the then refined it on DukeMTMC-reID based on two-stepped transfer learning approach. By freezing the base layers initially and unfreezing them after pre-training the randomly initialized layers, we preserved the model's performance while transitioning between datasets smoothly.

Our experiments showed that the refined OSNet significantly improved on DukeMTMC-reID while maintaining strong performance on Market1501. This demonstrated new model's ability to address diverse challenges in person re-ID and its effectiveness in complex real-world applications.

We will leverage the YOLOv8 [11] (You Only Look Once) object detection model in the application part. YOLOv8 is chosen for its state-of-the-art performance in terms of speed and accuracy, making it suitable for the real-time re-ID task, which requires high precision and fast processing.

### 1.2. Application

Our refined model was then applied to a video sequence that captures pedestrians passing by. Given a query image of a pedestrian, the target for our application is to identify all frames where the queried person appears, detect and crop out the person requested. This task can be applied in various areas of modern world automation, such as tracking persons of interest through security cameras, identifying particular pedestrians in autonomous driving, etc.

In this paper, we build out this application based on

an object detection model, such as YOLO [11], and a re-identifying network based on OSNet [21] trained on the Market1501 Dataset [16]. We also compare through real world examples on how the performance of our application has changed after applying transfer learning based on the Duke MTMC Dataset [12].

The video sample we have used to test our application is available here.

Figure 1: A frame from the video sample used in this paper. The scene features a fixed camera capturing pedestrians on a walkway in daylight, providing a suitable environment for testing person re-id models.

## 2. Related Work

As mentioned before, Person RE-ID faces numerous challenges from varying environment and similar appearances among individuals [16]. Traditional methods [5] [15]relied on appearance features for identification [2], but these struggled with variations in real-world settings.

Deep learning methods[13][6][18], like He et al. [7], used deep residual learning to address the vanishing gradient problem, providing strong capability of feature extraction but requiring high computational and memory resources. Howard et al. [9] introduced MobileNets, leveraging Depthwise Separable Convolutions for efficiency, making them suitable for real-time applications, though sometimes sacrificing accuracy as a cost. Chang et al. [1] proposed multi-level factorisation nets to handle multi-scale person images, improving robustness to viewpoint and distance variations but increasing model complexity and requiring extensive training data.

As a novel deep re-ID CNN, omni-scale network (OSNet)[21] [22] attempts to combines the advantages of these models, and is characterized by its effective implementation of omni-scale feature learning as its name suggests. Omni-scale means combining variable homogeneous and heterogeneous scales, each of which is composed of a mixture of multiple scales. The introduction of omni-scale boosts the performance of person re-ID because features corresponding small local regions (ie. clothes logo, hair style) and global whole body regions(ie. young women + white dress) are equally important in indicating the true match and imposter.

Based on OSnet's proven effectness in related task, it is selected as a foundation for this project, in which it is further refined to accommodate a broader datasets. More details about architecture and characteristics of OSnet as well as refining techniques will be revealed in the section of Method.

## 3. Method

### 3.1. Introduction to Existing Model: OSNet

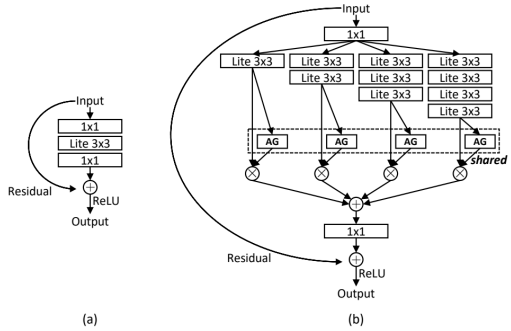


Figure 2: (a) Baseline bottleneck. (b) Proposed bottleneck. AG: Aggregation Gate. The first/last $1 \times 1$ layers are used to reduce/restore feature dimension.[21]

Different from conventional CNN which struggles to capture different spatial scales, OSnet is achieving excellent omni-scale feature learning by virtue of aggregation gate and omni-scale residual block. An aggregation gate is a mechanism used to manage how information flow from different sources is combined. It ensures the network is effectively aggregates or augment relevant information while filtering out noise or less important data. OSNet is implementing AG proposed by "Network In Network" (NIN) [10] combined with MLP and activation. On the other hand, residual block is the foundamental component in Residual Network (ResNet)[7]. Introduction of shortcut connections greatly mitigate the mitigate the vanishing gradient problem.

As is demonstrated in Figure 2, the proposed bottleneck in OSNet combined the ideas of AG and residual block. Note that Lightweight network designs(Lite) is used, and to be more specific, this architecture applies Depthwise Separable Convolutions[9], which drastically reduce the number of parameters and the computational burden.

All this features enables OSnet to specialize in learning omni-scale feature representations, making it well-suited for the person re-ID task.

### 3.2. Refinement of OSNet based on Two-stepped Transfer Learning

Though OSNet demonstrates excellent R1 and mAP as is illustrated in the figure in Appendices part, its performance degrades dramatically when tested on a dataset different from the training dataset. From our experiment, OSNet trained on Market1501 only gives an R1 of $30.1\%$ and mAP of $15.6\%$ when tested on DukeMTMC-reID.

A possible remedy is to combine all required dataset in the training stage, but it can leads to extremely high computational load and time-intensive training. Here, we are applying a technique named Two-stepped Transfer Learning[4]. It assumes we have pre-trained model trained either on a known or unknown dataset, in order to maintain the model's performance on a new dataset while avoid complete re-training, one can apply the strategy of two-step training.

The basic idea is to 'freeze' the base layers in the first segment of training process for few epochs(fixed base epochs), and only unfreeze them after the pre-training on the randomly initialized layers, which typically at the end of neural network, is complete. Such a learning strategy effectively avoid pre-tained model from being "polluted" by unfavorable back-gradient propagation as a consequence of randomly reinitialized layers and enabling a smooth transition from one dataset to the other.

| stage | output | OSNet |
|---|---|---|
| conv1 | 128×64, 64<br>64×32, 64 | 7×7 conv, stride 2<br>3×3 max pool, stride 2 |
| conv2 | 64×32, 256 | bottleneck × 2 |
| transition | 64×32, 256<br>32×16, 256 | 1×1 conv<br>2×2 average pool, stride 2 |
| conv3 | 32×16, 384 | bottleneck × 2 |
| transition | 32×16, 384<br>16×8, 384 | 1×1 conv<br>2×2 average pool, stride 2 |
| conv4 | 16×8, 512 | bottleneck × 2 |
| conv5 | 16×8, 512 | 1×1 conv |
| gap | 1×1, 512 | global average pool |
| fc | 1×1, 512 | fc |
| # params | | 2.2M |
| Mult-Adds | | 978.9M |

Figure 3: Architecture of OSNet with input image size $256 \times 128$ [21]

Shown in Figure 3. is an example architecture of OSNet with input image size $256 \times 128$. The randomly initialized layers, or the classifier, we want to pre-trained is the fc layer at the end of the network. We will experiment with varying numbers of fixed base epochs and maximum epochs to refine the model, ensuring it performs well on both datasets.

### 3.3. Application

Given the video sequence as input, our application processes the sequence frame-by-frame. On each iteration, we firstly apply YOLO [11] provided by Ultralytics https://github.com/ultralytics/ultralytics to detect all objects labeled as "person". The detected bounding boxes are then resized to 16-by-64, so that we can work with the detected persons on a unified scale. Accordingly, the input query image that contains the target person is also re-scaled to the same dimensions.

The trained OSNet is then applied to both the query image and each detected bounding box. When an image goes through OSNet, a flattened feature vector is produced by the underlying neural network. Feature vectors extracted from both the query image and the candidate detection are then normalized. After that, a distance metric between the two normalized feature vectors are computed. In our application, we chose

$$\text{distance} = 1 - \cos\theta = 1 - a \cdot b$$

where $a$ and $b$ are normalized features as a measurement of similarity. As a result, the computed distance should be between 0 to 2, and a smaller distance represents higher similarity. If the distance metric is below a certain threshold, meaning the similarity between the two are small, we conclude that the candidate bounding box contains the same person in the query image. Our algorithm is illustrated in 1.

**Algorithm 1:** Identify person queried in the current frame

**Input:** frame, query
**Output:** detection
**Input :** Picture frame, Query image
**Output:** The same person in the query image that appears in the frame

```
queryFeature ← OSNet(query);
queryFeature ← Normalize(queryFeature);
personBoxes ← YOLO(frame);
foreach person in personBoxes do
    personFeature ← OSNet(person);
    personFeature ← Normalize(personFeature);
    distance ← queryFeature · personFeature;
    if distance < threshold then
        detection ← person;
        break;
return detection
```

## 4. Dataset and Features

We are utilizing two datasets in this project: Market-1501[16] and DukeMTMC-reID[18]. See sample images

Figure 4: The images depict pedestrians captured from various camera angles within the Market-1501 dataset[16]. In each pair of adjacent images, the pedestrians shown belong to the same identity.

Figure 5: The images include sample pedestrians from the DukeMTMC-reID dataset [18], where each set of adjacent images in both rows and every five columns represents the same identity.

Figure 4 and Figure 5

The Market-1501 dataset, introduced in 2015, is a widely-used benchmark for person re-ID tasks. It contains over 32,000 images of 1,501 identities captured by 6 cameras. The dataset includes images with varied resolutions, typically around 128x64 pixels, all annotated with unique identity labels. Challenges in this dataset lie in varying illumination, occlusions, different poses, and background clutter. It is divided into a training set with 12,936 images of 751 identities and a testing set with 19,732 images of 750 identities.

The DukeMTMC-reID dataset is a subset of the DukeMTMC multi-target tracking dataset, and it is tailored for re-ID tasks. It consists of more than 36,000 images of 1,404 identities, captured by 8 cameras. Images resolution are also around 128x64 pixels and annotated with unique identities and camera IDs. The dataset features diverse environments and complex backgrounds. The training set includes 16,522 images of 702 identities, with the testing set comprising 17,661 images of 702 identities.

The DukeMTMC-reID dataset presents a higher level of difficulty compared to Market-1501. This increased difficulty arises from DukeMTMC-reID's more diverse scenes and complex backgrounds, captured by a larger number of cameras, as mentioned before. These factors introduce greater variability in viewpoints, lighting conditions, and occlusions, making the re-ID task more challenging. On the other hand, Market-1501 offers a relatively uniform environment, resulting in slightly less variation across images.

We will see in subsequent section that when model trained on Market-1501 is tested on DukeMTMC-reID, signficant decline in performance is spot, and we will introduce efficient methods to accommodate the pre-trained model to DukeMTMC-reID, ensuring small and acceptable training time. Note data augmentation such as Random flip and color jitter are applied during training stage.

## 5. Results Discussion

### 5.1. Refinement Process

| Fixed Epoch | Pre-Transition Accuracy | Post-Transition Accuracy | Accuracy Drop |
|---|---|---|---|
| 10 | 90.4883 | 49.3750 | 41.1133 |
| 20 | 90.6250 | 71.8750 | 18.7500 |
| 30 | 95.3125 | 79.6875 | 15.6250 |
| 35 | 95.8464 | 84.4531 | 11.3933 |
| 50 | 93.7500 | 40.6250 | 53.1250 |

Table 1: Impact of Fixed Epochs on Accuracy During Transition Stage. This table presents the pre-transition and post-transition accuracies for different fixed epochs during two-stepped transfer learning. The accuracy difference highlights the drop observed at the transient stage from frozen to unfrozen layers.

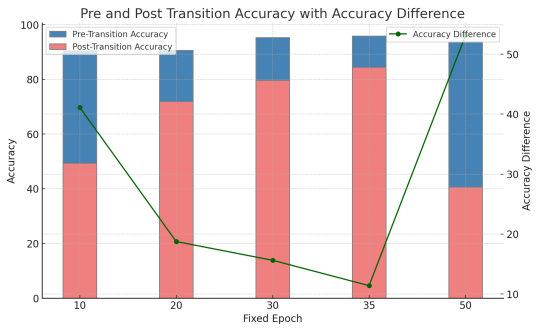


Figure 6: Comparison of Pre-Transition and Post-Transition Accuracies Across Different Fixed Epochs in Fine-Tuning Stage. The bars represent the accuracy before (steel blue) and after (light coral) unfreezing the base layers. The green line indicates the accuracy difference, highlighting the impact of insufficient adaptation when the base layers are fixed for an extended period

An important parameter to determine in Two-Stepped Transfer Learning is the number of fixed epochs. By selecting an appropriate number of fixed epochs, we can en-

sure a controlled transition from regional training to global training, minimizing performance drop during the process.

As is illustrated in Table 1, accuracy difference start to mitigate as we gradually increase fixed epoch from 10. We obtains minimal performance degradation when then fixed epoch is set to 35, where the model only experiences an training accuracy drop of $11.3933\%$. And as the fixed epoch goes further larger, we observe a drastic decline when transition happens, possibly as a result of insufficient adaptation of the base layers. As we fixed the extended layers for an extended epochs, and suddenly unfreeze it, the base layers have to make significant adjustments to accommodate to the new dataset. Since they were fixed for too long, they may struggle to quickly adapt to the new data, leading to sharp drop, as is observed when fixed epoch is increased to 50, as is shown in Figure 6.

In addition to the number of fixed epoch, number of maximal epoch should also be carefully tuned. Note this typically differs from the epoch number used on training a new model. Since we are attempting to fine-tune a pre-trained model on a different dataset, we hope the final model to perform well on both datasets with minimal bias towards either one. Therefore, at the stage of fine-tuning, it is a common to use a max epoch smaller than that in common training. For OSNet, the training duration is approximately 250, while for the fine-tuning, it is usual to select an epoch number much smaller.

The maximal epoch chosen in this project is 50, Note that fixed epoch is counted into maximal epoch. So for the refinement, the base network will be fixed for 35 epochs and then open for training for 15 epochs, while the classifier is active throughout the whole process. We choose a relatively low maximal epoch because we don't hope OSNet fits itself too much to DukeMTMC-reID such that it 'forgets' what is learned in previous dataset. Also, a lower epoch means less computational load and less extended training time, which is always preferred in the process of refinement.

## 5.2. Refined Model Analysis

The origin OSNet trianed on Market1501 demonstrates on outstanding performance when tested on the same dataset, mAP reaches $85.7\%$, Rank-1 is $94.6\%$, and Rank-20 achieves $99.2\%$. However, when being tested on DukeMTMC-reID, its accuracy undergoes a dramatic drop: the mAP becomes $15.6\%$, Rank-1 is $44.0\%$, and even the Rank-20 is only $56.4\%$. It is obvious that OSNet struggles when exposed to dataset never seen before, even though the setting are similar.

In contrast, after OSNet is refined on DukeMTMC-reID, though we see an expected decreasing performance on Market1501(mAP: $56.9\%$, Rank-1: $80.8\%$, Rank-20: $96.6\%$), the new model's evaluation result on DukeMTMC-reID soars(mAP: $58.3\%$, Rank-1: $78.2\%$, Rank-20: $93.2\%$). See

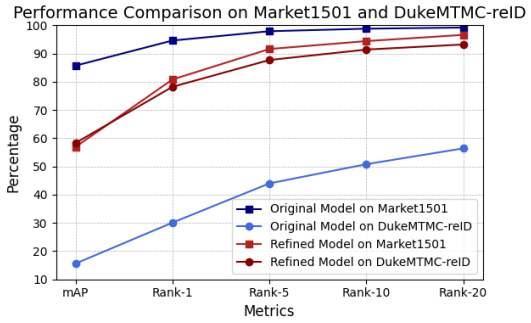


Figure 7: The graph compares the performance of original and refined models on Market1501 and DukeMTMC-reID datasets using metrics mAP, Rank-1, Rank-5, Rank-10, and Rank-20. Navy blue (Market1501) and royal blue (DukeMTMC-reID) lines represent the original model, while firebrick (Market1501) and dark red (DukeMTMC-reID) lines represent the refined model. The refined model significantly improves on DukeMTMC-reID and achieves balanced performance across both datasets. The x-axis shows the metrics, and the y-axis indicates their percentage values.

Figure 7. This means new model process the capability of person re-ID in the settings of both datasets, and the overall Rank-1 accuracy is around $80\%$, and the average Rank-2 accuracy achieves $95\%$.

## 5.3. Application

With our application built out, we applied the YOLO model [11] for person detection and two OSNet models [21], one trained with the Market1501 dataset [16] (referred to as the original model) and one obtained through transfer learning on DukeMTMC-reID dataset [12] (referred to as the refined model) for feature extraction. Our application takes a video sample and a query person as input. It detects each person appearing in the video sequence frame-by-frame, extracting and comparing the features of persons of interest and persons detected, and outputs all occurrences of the queried person in the given video sequence.

In our experiments, we query for multiple persons (See Figure 8) of different dressing colors and compare the performance of the original model and the refined model. Since our refined model incorporates both datasets, we expect it to have better performance.

The query targets we choose are

- Query A, a pedestrian in red.
- Query B, a pedestrian in yellow.
- Query C, a pedestrian in light blue.
- Query D, a pedestrian with a white bag.

Query A Query B Query C Query D

Figure 8: Queried pedestrian targets.

Our application is quite successful in detecting the queried individuals appearing in the sample video. Both the original model and the refined model delivered good performance in detecting all frames where the queried target appears, with an acceptable failure rate of false positives. Below we demo a selection of the results on the sample video with the specified query targets mentioned above. In the demo, the application employs the refined model. The distance threshold was set at 0.20.

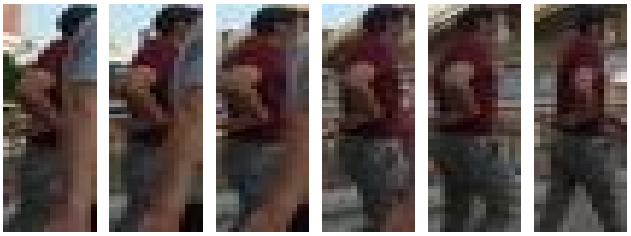

Figure 9: Pedestrian in red (query A) detected in the video sample.

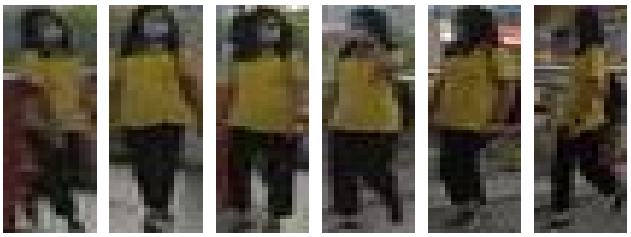

Figure 10: Pedestrian in yellow (query B) detected in the video sample.

As shown in the demo results displayed in Figure 9, 10, 11, 12 our application was successful in detecting and extracting the queried target, even when the targets appear in the video sequence at different positions or orientations, with various gestures, and with pictures taken at different angles. Our application was also robust in handling constantly changing backgrounds and certain occlusions.

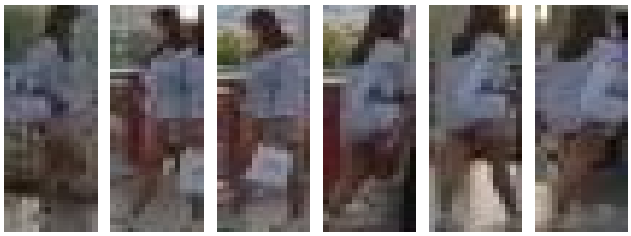

Figure 11: Pedestrian in blue (query C) detected in the video sample.

Figure 12: Pedestrian with a white bag (query D) detected in the video sample.

### 5.4. Statistics and Analysis

In this section, we discuss the statistics collected from applications based on the original model and the refined model, and compare their performance.

The goal of this application is to detect and extract the queried person from all frames where they appear in the video sequence. Therefore, we collect the number of frames where the target was successfully detected, and compare it with the actual number of frames where the target appears. The rate of successful detections should be approaching 100% with higher performance. Besides that, we must also consider the failure cases where the wrong target was detected and extracted. This happens when the model outputs a candidate that is actually not the person of interest, usually due to similar dressing, positions, or gestures are present on different persons. Such cases are considered to be ”false positives”, with the number closer to 0 with higher performance. Here we demo some examples of failure cases. See Figure 13

These false positive cases happened when querying for the target ”pedestrian in yellow (query B)”. We can see that although these pedestrians returned by the application are not the same person as the queried target, they have many similar appearances, such as yellow t-shirt, black trousers, black hair, etc. These similar attributes might have caused the feature vectors to present a smaller distance.

Some important indicators of performance for this application include:

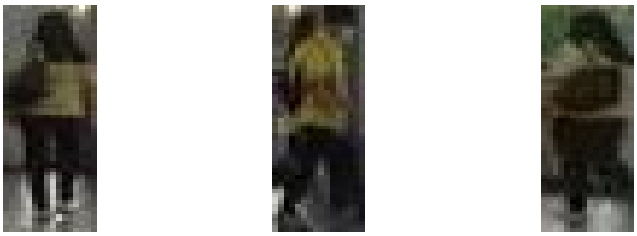

Figure 13: False positive cases for the pedestrian in yellow (query B) by the original model.

- Rate of successful detections.
- Number of false positives.

We collected detection statistics for the four queries on the original model and the refined model, with distance threshold set at 0.20.

| Query | Detections | Total Target Appearances | Rate of Success | Number of False Positives |
|---|---|---|---|---|
| A | 6 | 16 | 38% | 0 |
| B | 99 | 244 | 41% | 5 |
| C | 182 | 318 | 57% | 0 |
| D | 42 | 49 | 86% | 0 |

Table 2: Detection Results Using the Original Model

| Query | Detections | Total Target Appearances | Rate of Success | Number of False Positives |
|---|---|---|---|---|
| A | 9 | 16 | 56% | 0 |
| B | 201 | 244 | 82% | 2 |
| C | 163 | 318 | 51% | 0 |
| D | 37 | 49 | 76% | 0 |

Table 3: Detection Results Using the Refined Model

As summarized in the results table, our refined model has gained a significant performance increase for query A and query B, with rate of success increasing by 18% and 41%, respectively. While the refined model's performance slightly decreased for query C and query D, the decrease was much less significant compared with the marginal gain for the other two queries. These results are in line with our expectations, since the refined model incorporated training from both the Market1501 dataset [16] the DukeMTMC-reID dataset [12], which made the model more sensitive and robust, especially on brighter colors. In particular, the refined model exhibited much better capability in detecting and distinguishing the pedestrian in query B from other similar dressing persons. These results on the application also solidify the evaluation gains we saw from the models part.

## 6. Conclusion/Future Work

We studied the problem of person re-identification in a video sequence. We explored a novel deep learning based model, the OSNet [21] [22], and trained the network on the Market1501 Dataset [16]. We then applied transfer learning on the Duke MTMC Dataset [12] and obtained a refined model. We discovered that the refined model gained significantly higher evaluation performance on the Duke MTMC Dataset, while limiting the accuracy drop on the Market1501 Dataset.

Using the model's feature extractors, we built a robust application that takes any video sequence and a query as input, and outputs all occurrences of the target query in the video sequence. We then compared the performance of the original and the refined model on our application with real-world examples. We discovered that the refined model achieved significantly higher detection rates for certain query targets, while limiting the detection rates drop for some queries, solidifying the evaluation gains on the Market1501 and Duke MTMC datasets.

For future work, we could explore more techniques for further improving the model, such as trying the triplet loss [8] during training. In recent years, person re-identification in the 3D space is also becoming more and more popular [17], where unsupervised 3D reconstruction methods are employed [19] in novel 3D models. With the rise of the transformer architecture and large language models, person re-identification could also be formulated as a text-based retrieval problem [14]. Like any computer vision task, the topic of person re-identification is a fast-evolving and exciting task in contemporary AI.

## 7. Appendices

| Method | Publication | Backbone | Market1501 | | CUHK03 | | Duke | | MSMT17 | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | R1 | mAP | R1 | mAP | R1 | mAP | R1 | mAP |
| ShuffleNet†‡ [78] | CVPR'18 | ShuffleNet | 84.8 | 65.0 | 38.4 | 37.2 | 71.6 | 49.9 | 41.5 | 19.9 |
| MobileNetV2†‡ [43] | CVPR'18 | MobileNetV2 | 87.0 | 69.5 | 46.5 | 46.0 | 75.2 | 55.8 | 50.9 | 27.0 |
| BraidNet† [63] | CVPR'18 | BraidNet | 83.7 | 69.5 | - | - | 76.4 | 59.5 | - | - |
| HAN† [29] | CVPR'18 | Inception | 91.2 | 75.7 | 41.7 | 38.6 | 80.5 | 63.8 | - | - |
| OSNet† (ours) | ICCV'19 | OSNet | 93.6 | 81.0 | 57.1 | 54.2 | 84.7 | 68.6 | 71.0 | 43.3 |
| DaRe [64] | CVPR'18 | DenseNet | 89.0 | 76.0 | 63.3 | 59.0 | 80.2 | 64.5 | - | - |
| PNGAN [39] | ECCV'18 | ResNet | 89.4 | 72.6 | - | - | 73.6 | 53.2 | - | - |
| KPM [46] | CVPR'18 | ResNet | 90.1 | 75.3 | - | - | 80.3 | 63.2 | - | - |
| MLFN [2] | CVPR'18 | ResNeXt | 90.0 | 74.3 | 52.8 | 47.8 | 81.0 | 62.8 | - | - |
| FDGAN [11] | NeurIPS'18 | ResNet | 90.5 | 77.7 | - | - | 80.0 | 64.5 | - | - |
| DuATM [47] | CVPR'18 | DenseNet | 91.4 | 76.6 | - | - | 81.8 | 64.6 | - | - |
| Bilinear [52] | ECCV'18 | Inception | 91.7 | 79.6 | - | - | 84.4 | 69.3 | - | - |
| G2G [44] | CVPR'18 | ResNet | 92.7 | 82.5 | - | - | 80.7 | 66.4 | - | - |
| DeepCRF [3] | CVPR'18 | ResNet | 93.5 | 81.6 | - | - | 84.9 | 69.5 | - | - |
| PCB [53] | ECCV'18 | ResNet | 93.8 | 81.6 | 63.7 | 57.5 | 83.3 | 69.2 | 68.2 | 40.4 |
| SGGNN [45] | ECCV'18 | ResNet | 92.3 | 82.8 | - | - | 81.1 | 68.2 | - | - |
| Mancs [60] | ECCV'18 | ResNet | 93.1 | 82.3 | 65.5 | 60.5 | 84.9 | 71.8 | - | - |
| AANet [56] | CVPR'19 | ResNet | 93.9 | 83.4 | - | - | 87.7 | 74.3 | - | - |
| CAMA [71] | CVPR'19 | ResNet | 94.7 | 84.5 | 66.6 | 64.2 | 85.8 | 72.9 | - | - |
| IANet [17] | CVPR'19 | ResNet | 94.4 | 83.1 | - | - | 87.1 | 73.4 | 75.5 | 46.8 |
| DGNet [84] | CVPR'19 | ResNet | 94.8 | 86.0 | - | - | 86.6 | 74.8 | 77.2 | 52.3 |
| OSNet (ours) | ICCV'19 | OSNet | 94.8 | 84.9 | 72.3 | 67.8 | 88.6 | 73.5 | 78.7 | 52.9 |

Figure 14: Results (surpassing most published methods by a clear margin. It is noteworthy that OSNet has only 2.2 million parameters, which are far less than the current best-performing ResNet-based methods. -: not available. †: model trained from scratch. ‡:reproduced by us. (Best and second best results in red and blue respectively) [21]

## 8. Contributions and Acknowledgements

Both authors have made equal contributions towards this work. Zhiyuan Li was mainly responsible for the network training and refining part, and Jiayang Wang was mainly responsible for building out the application for real world examples and evaluations.

Throughout our project, we trained our models and built the application on top of Torchreid [20], the code base for the original research on OSNet [21] [22] https://github.com/KaiyangZhou/deep-person-reid. We are grateful to the authors for publishing this wonderful work.

When building out the application, we used the YOLO object detection network [11] from Ultralytics https://github.com/ultralytics/ultralytics, a high-performing object detection model.

We have trained our model on the Market1501 Dataset [16] and the Duke MTMC Dataset [12]. In our evaluations, we used a video sample with pedestrians passing by published on Pixabay, https://pixabay.com/videos/pedestrian-asia-thailand-bangkok-113423/, filmed by an author under the name of "viarami", a photographer from Germany.

We utililized Google Cloud Platform's compute instance with NVIDIA Tesla T4 GPU for training and evaluations. We are grateful for the generous cloud credits from Google and the CS231N course.

## 9. References/Bibliography

## References

[1] X. Chang, T. M. Hospedales, and T. Xiang. Multi-level factorisation net for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2109–2118, 2018.

[2] Y.-C. Chen, W.-S. Zheng, J.-H. Lai, and P. C. Yuen. An asymmetric distance model for cross-view feature mapping in person re-identification. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(8):1661–1675, 2016.

[3] M. Farenzena, L. Bazzani, A. Perina, V. Murino, and M. Cristani. Person re-identification by symmetry-driven accumulation of local features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2360–2367, 2010.

[4] M. Geng, Y. Wang, T. Xiang, and Y. Tian. Deep transfer learning for person re-identification. *2018 IEEE Fourth International Conference on Multimedia Big Data (BigMM)*, pages 1–5, 2016.

[5] D. Gray and H. Tao. Viewpoint invariant pedestrian recognition with an ensemble of localized features. In *European Conference on Computer Vision*, pages 262–275. Springer, Berlin, Heidelberg, 2008.

[6] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[7] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2015.

[8] A. Hermans, L. Beyer, and B. Leibe. In defense of the triplet loss for person re-identification. *ArXiv*, abs/1703.07737, 2017.

[9] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications, 2017.

[10] M. Lin, Q. Chen, and S. Yan. Network in network. *CoRR*, abs/1312.4400, 2013.

[11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[12] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *European Conference on Computer Vision (ECCV)*, 2016.

[13] Y. Sun, L. Zheng, Y. Yang, Q. Tian, and S. Wang. Beyond part models: Person retrieval with refined part pooling (and a strong convolutional baseline). In *Proceedings of the European conference on computer vision (ECCV)*, pages 480–496, 2018.

[14] S. Yang, Y. Zhou, Y. Wang, Y. Wu, L. Zhu, and Z. Zheng. Towards unified text-based person retrieval: A large-scale multi-attribute and language search benchmark. In *Proceedings of the 2023 ACM on Multimedia Conference*, 2023.

[15] R. Zhao, W. Ouyang, X. Wang, and X. Li. Unsupervised salience learning for person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3586–3593, 2013.

[16] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian. Scalable person re-identification: A benchmark. In *Computer Vision, IEEE International Conference on*, 2015.

[17] Z. Zheng, X. Wang, N. Zheng, and Y. Yang. Parameter-efficient person re-identification in the 3d space. *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, 2022. doi:10.1109/TNNLS.2022.3214834.

[18] Z. Zheng, L. Zheng, and Y. Yang. Unlabeled samples generated by gan improve the person re-identification baseline in vitro, 2017.

[19] Z. Zheng, J. Zhu, W. Ji, Y. Yang, and T.-S. Chua. 3d magic mirror: Clothing reconstruction from a single image via a causal perspective. *arXiv preprint arXiv:2204.13096*, 2022.

[20] K. Zhou and T. Xiang. Torchreid: A library for deep learning person re-identification in pytorch. *arXiv preprint arXiv:1910.10093*, 2019.

[21] K. Zhou, Y. Yang, A. Cavallaro, and T. Xiang. Omni-scale feature learning for person re-identification. In *ICCV*, 2019.

[22] K. Zhou, Y. Yang, A. Cavallaro, and T. Xiang. Learning generalisable omni-scale representations for person re-identification. *TPAMI*, 2021.