Fast and Lightweight Online Person Search for Large-Scale Surveillance Systems

Andreas Specker[3,1] Lennart Moritz[1,2] Mickael Cormier[3,1] Jürgen Beyerer[1,3]

[1]*Fraunhofer IOSB*, Karlsruhe, Germany; [2]*Fraunhofer Center for Machine Learning*;
[3]*Vision and Fusion Lab, Institute for Anthropomatics and Robotics*,
Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany

firstname.lastname@iosb.fraunhofer.de

Abstract

The demand for methods for video analysis in the field of surveillance technology is rapidly growing due to the increasing amount of surveillance footage available. Intelligent methods for surveillance software offer numerous possibilities to support police investigations and crime prevention. This includes the integration of video processing pipelines for tasks such as detection of graffiti, suspicious luggage, or intruders. Another important surveillance task is the semi-automated search for specific persons-of-interest within a camera network. In this work, we identify the major obstacles for the development of person search systems as the real-time processing capability on affordable hardware and the performance gap of person detection and re-identification methods on unseen target domain data. In addition, we demonstrate the current potential of intelligent online person search by developing a real-world, large-scale surveillance system. An extensive evaluation is provided for person detection, tracking, and re-identification components on affordable hardware setups, for which the whole system achieves real-time processing up to 76 FPS.

1. Introduction

Image and video data acquired by surveillance cameras is an indispensable asset in the fight against crime and terrorism. Therefore, the number of surveillance cameras continues to grow rapidly, despite public concerns about data privacy. A city-scale video surveillance camera network quickly results in large amounts of video data that can no longer be manually analyzed by video investigators. Sifting through video footage is a strenuous task for human operators, requiring a high degree of concentration at all times, therefore leading to an increased potential for mistakes over time. It is especially the case when several camera streams have to be monitored simultaneously.

Intelligent video surveillance is an active research field that mitigates these problems by providing algorithm-based assistance to system operators or automating specific tasks completely. Assisting tools may support the recognition of suspicious activities [28, 43], the re-identification (re-id) [55, 64] or tracking [3, 36, 52] of persons visible in the camera network. A comprehensive framework should be able to automatically detect a person involved in an assault using activity recognition algorithms and notify authorities. Subsequently, security personal can use person search methods to track the perpetrators' escape routes and arrest them. In this work, we focus on the search part and design, implement, and thoroughly evaluate a system allowing the online search of all occurrences of a person-of-interest in a real-world camera network. This system is able to process incoming video data in real-time to allow searching the current footage without delay or collecting further evidence after an incident has happened. Mitigation strategies are proposed and evaluated to improve results in person detection, tracking, and re-id in real-world data (see Figure 1) and their associated challenges. Furthermore, the run-time of each method is improved to enable the whole system to run in real-time on affordable consumer hardware. The effectiveness of the proposed approaches is proven using video frames acquired by real-world surveil-

Figure 1: Real-world camera view. The scene is cluttered, contains small persons, and many occluded areas.

lance cameras. In summary, our contribution is threefold: (1) we propose a collection of lightweight individual modules for person detection, tracking, and re-id in real-world settings and prove their efficiency; (2) we design and implement an entire system for online person search and address several challenges related to real-world deployment; (3) finally, we prove the real-time capability of our system and achieve up to 76 frames per second (FPS) on affordable consumer hardware and therefore offer realistic baselines for real-world applications.

2. Related Work

Person Detection With the increasing success of deep learning, several public datasets emerged which focus on pedestrians in real-world conditions [10, 24, 34, 42, 49, 58] as well as realistic simulated worlds [12, 13]. The availability of such datasets facilitated the adaptation of meta-architectures, *e.g*. SSD [30], YOLO [5, 37] and Faster R-CNN [38], to the task of pedestrian detection [57]. There are hereby different approaches to the task, such as using instance segmentation [14, 33] or points [11, 54] for detection guidance. Recent approaches focus specifically on pedestrian detection in crowded scenes [16, 29, 31, 67] and their specific challenges such as occlusion [59] and detection of person at small-scale [21]. However, such approaches mainly address improvement towards better accuracy and do not consider run-time improvements. Model speed improvements are achieved through specially designed lightweight architectures [9, 47, 53] or quantization [6, 8] which both improve the run-time and memory footprint of the model with acceptable drop in accuracy.

Single-camera Tracking Similar to this work, a majority of single-camera tracking algorithms build on the tracking-by-detection paradigm. Following this procedure, the single-camera tracking is divided into the detection and the following track association stages. An important clue to whether a detection belongs to an already existing track is delivered by the position in the camera frame. Bochinski et al. [3] solely rely on the position in their association procedure by computing the overlap between unassigned detections and existing tracks. While this approach delivers fast results due to reduced complexity, the quality of the results drops for occlusion. To tackle this problem, the SORT tracker [2] incorporates a Kalman filter with a linear motion model to predict and use the motion direction and velocity of people for the association. Further improvements include the tracking of overlapping objects and enabling the continuation of tracks after mid-term whole-body occlusions based on visual features [4, 50, 52]. More recent works integrate the track association step into the detector [1, 66] or build on 3D CNNs [36], which process videos and output whole tracks instead of detections. For a real-time capable processing pipeline, simple methods such as enhanced versions of the SORT [2] tracker are often preferred, as in this work.

Person Re-identification A variety of datasets have been published in recent years that mimic real-world scenarios for person re-id [22, 23, 25, 39, 51, 62]. However, complex state-of-the-art architectures tend to overfit training data. Therefore, research efforts focus on generalizable methods which either aim at learning robust visual features from a single dataset [20, 26, 45, 65] or a combination of multiple datasets [7, 19, 44, 48, 61]. Due to the availability of appropriate datasets, we rely on the second approach. Recent works apply data augmentation strategies [48], introduce a separate mapping network [44], or integrate either instance [7, 19] or batch [61] normalization layers. However, an additional network increases computation time, and some normalization layers tend to omit relevant information. As a result, we focus on baseline approaches and established best practices [32, 56] in conjunction with several data augmentation strategies [35, 63].

3. Background and Motivation

Person Search is the task of finding all occurrences of a person-of-interest in a database containing large amounts of image or video data. The starting point of a search is usually an image of the person [17]. Nonetheless, recent works consider input from textual descriptions of a person's semantic attributes [18, 46]. In general, person search is the combination of person detection and re-id. To this aim, whole images or video frames are used, which differs from person re-id, for which person crops are typically processed. Person search is typically considered a semi-automatic approach since the operator provides the probe image. A person re-id CNN extracts feature vectors for this query image and each detected person in the gallery database. Afterward, the similarity in visual appearance is measured by computing a distance metric between the feature vectors. As a result, a ranked list of persons in the gallery sorted by their distance to the query feature vector is returned. Over time multiple entries of a single target are added to the gallery. However, the contribution of such duplicated entries is limited since being close in time, location, and from the same camera. Therefore, we additionally incorporate a single-camera tracking step after detection. It clusters occurrences of persons within a camera and thus improves the rankings. Furthermore, aggregating feature over multiple time steps leads to more robust features for person search. Momentary occlusion of body parts loses in influence, and representations of people within a track are improved through views from multiple camera angles.

Real-World Deployment of person search is a challenging problem regarding several aspects. While current state-of-the-art methods for person detection, tracking, and re-id indicate high and near-perfect results on benchmark datasets, most approaches seem to overfit the distributions of those

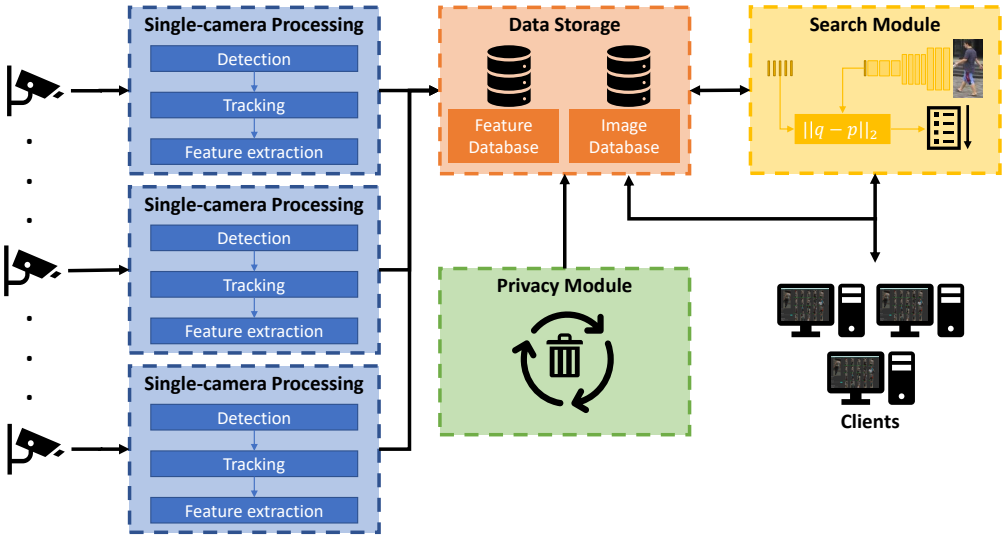

Figure 2: Overview of the modules of our system. Colored frames illustrate the different Docker containers. Each camera stream is processed in an instance of the single-camera processing container (blue), which sends the extracted data to the data storage container (orange). The privacy and the search module interact with the data storage container to process search requests and to comply with legal requirements. To visualize search results, clients have access to the image database.

datasets. Therefore, different luminosity, brightness, or view angle are often enough to confuse overfitted models. As a result, great computational efforts are required for disappointing results due to poor generalization. For instance, an independent study reported that the recognition accuracy of the face recognition system installed in London over a large number of cameras was below 20% in the period studied [41]. However, public safety agencies, such as police departments, are often severely limited in financial resources. Therefore, such a system is required to perform with affordable hardware. Nevertheless, the processing of surveillance video streams is required in real-time to ensure rapid intervention after an incident. Searching for persons should be possible in current video frames without delay to track the movements of possible suspects. Furthermore, to ensure admissibility of re-id results in a court of justice, a person search system is required to deliver high re-id accuracy in real-world data at any time of the day, in any season or weather. Moreover, a real-world system requires flexibility and scalability. In the case of mass gathering events and states of high alert, supplementary mobile cameras are often used in addition to fixed and static surveillance cameras. Thus, it should be possible to include these new sources in the re-id system without impairing the overall performance. Similarly, it must be possible to connect additional servers or databases without interruption of the services. Finally, strict laws regarding data protection are implemented in most countries. Often only short-term storage of footage is allowed on ring-buffer memory. Hence, the system needs to comply with these regulations and implement appropriate mechanisms in all layers.

4. Real-world Person Search System

This section outlines the proposed real-world and real-time person re-id system. An overview of the complete system is provided in Section 4.1. Subsequently, the separate system modules are presented in detail.

4.1. Overview

Our system for real-world person search is designed for high flexibility and scalability. It consists of separate modules that communicate with each other and is illustrated in Figure 2. Video streams produced by surveillance cameras are the input for the single-camera processing component. Each video feed has its separate processing instance where persons are detected and tracked within the camera view. A re-id CNN generates a feature representation of the persons' whole-body appearances, as the basis for person retrieval. The extracted information is forwarded to the data storage module. It stores meta information of tracks, corresponding feature vectors, and image or video data for visualization purposes. The task of the privacy module is to ensure legal requirements. This module is con-

nected to the database and regularly reviews the age of the stored data. The search module receives search queries from clients, processes them, and returns a ranked list of tracks enriched with metadata. In Figure 2, each color represents a Docker container with standardized interfaces. Scaling up the system is achieved by simply connecting new servers to host more single-camera processing containers. Furthermore, adding or removing camera streams to the processing pipeline solely requires starting or stopping a Docker container with adjusted configuration parameters.

4.2. Single-camera Processing

The single-camera processing component analyzes the video streams of the surveillance cameras and creates tracks of persons with associated person embeddings from a re-id model. First, we rely on the YOLOv4 [5] architecture to detect the persons present in each frame, as it provides a favorable trade-off between speed and accuracy [9]. Afterward, detections are passed to the single-camera tracker. We implement the SORT [2] as well as the DeepSORT [52] tracking approach in our system. Common issues are dropped frames caused by a poor IP camera connection which impairs the performance of such trackers due to their motion-based design. We address this by assigning frame numbers to the incoming frames. When missing frames are detected, the tracker is fast-forwarded by feeding empty frames to maintain accurate motion estimates for the tracked objects. In contrast to SORT, person embeddings are used in DeepSORT for the association step, which support person re-id after occlusions. However, leveraging such embeddings requires distance computations in every update step and thus much more computation time. We address this problem by introducing a so-called Track Merging Module (TMM) that extends the SORT tracker. In this version, tracks terminated by the SORT tracker are not directly stored in the database and deleted by the track management. Instead, such tracks receive a new track state *pending*. When a new track is created by the SORT tracker, the TMM compares its visual embedding to the aggregated features of every *pending* track by computing the Euclidean distance to those (see Sec. 4.5). In case that the distance to a track is below a threshold $\delta_{TMM} = 0.25$, this track is treated as the continuation of the *pending* track, and the latter is re-opened. The threshold is adjustable to account for the different characteristics of scenarios to ensure an acceptable level of false positive matches. Tracks are otherwise closed and stored in the database after spending $p = 30$ subsequent frames in the *pending* state. Therefore, time-consuming person embedding comparisons are only occasionally necessary, and yet temporarily occluded tracks may be resumed.

To extract features for person re-id, we use the OSNet [64] network architecture trained on multiple datasets. Moreover, several optimizations are applied to improve the generalization ability of the trained model. A thorough description of design choices and extensive experimental evaluation is provided in Section 6.3. We chose this model since it offered the best compromise between generalization and speed in our preliminary experiments.

In case that the video processing module shows a delay against the incoming video stream, we apply a frame skipping strategy. When the FIFO buffer reaches its maximum size, the oldest entries are discarded and the tracker's movement predictions are adjusted as described earlier. In addition, we apply multiprocessing with the producer-consumer architecture to separate the reading and decoding operation of the video stream from the processing loop and speed up the computation. Furthermore, we rely on asynchronous calls to our database to prevent network and storage delays from producing processing overhead.

4.3. Data Storage

The data storage component only persists data for aggregated tracks instead of single detections. Three types of information have to be stored for each aggregated track.
Metadata: Contains information about the camera a track was captured on, start and end times, bounding box positions, and a reference to the associated image data.
Feature data: We only store one feature vector per track. This feature vector is the average of all extracted feature representations for one track. Storing only one aggregated feature saves storage space and speeds up computation time during the re-id task.
Image/video data: To display retrieval results in a meaningful way, we store images of the detected person within one track. Concurrent network access is required for the data storage in order to guarantee flexibility and scalability, *e.g*. the data storage is accessed from multiple servers processing multiple camera streams. Therefore, instead of a compact system, such as SQLite, we use Docker to host a MySQL server as a relational database management system (RDBMS), which is well-suited to the structure and size of our data. It offers fast access times, flexible SQL queries that allow for filtered searches, and distributed access over the network. Since video management solutions are typically already used by security authorities, separate video data storage is often not necessary. Therefore we intend for our system to be connected to a pre-existing video management system individually. For development, we use a network shared file system hosted on the same machine to mimic a pre-existing video management system, where our image data is stored.

4.4. Privacy Module

The privacy module ensures compliance with Art. 17 of the European Union's General Data Protection Regulation (GDPR) by enforcing maximum storage times for

personal data such as feature representations and images. Through a configurable maximum storage time, this module frequently checks the database for entries that exceed the storage time limit. It ensures data integrity by attempting to delete the referenced image data from the file system first and only deleting the meta and feature data entry along with the files system reference from the SQL database if the first step was carried out successfully. To prevent users from simply disabling the privacy module to bypass legal regulations, we store a timestamp of the most recent contact between our database and the privacy module in a secured database within our data storage module. We only allow our single-camera processing and search modules to access the database if the timestamp does not exceed the privacy module's update frequency.

4.5. Search Module

The task of the search module is to process incoming requests from client applications. It features a standardized interface for making search queries and transferring the results. Besides the query images, search requests contain the camera streams and time intervals to be searched. It is possible to conduct a typical person search or to provide a watch list of person images. In the first case, a ranked list of tracks included in the database sorted by their distance is created based on a query image. For this, the probe image is embedded into the visual feature space using the re-id CNN, and subsequently, the Euclidean distance is calculated to relevant tracks from the database to measure the similarity. Since we normalize the features beforehand, the same ranking accuracy is achieved as with the cosine distance. However, in our experiments, the calculation requires only two thirds of the runtime compared to the cosine distance. The metadata of the best matching track is returned to the requesting application. In the second case, query features are extracted analogous to the offline search. However, instead of comparing the feature vector to those stored in the database, only new database entries are considered. If the distance to a track is below an adjustable threshold, a notification with metadata associated with the track will be sent.

5. Implementation Details

We use configurable Docker containers to deploy and develop the system, allowing us to set up the components locally with only one or two cameras or on a distributed network of cameras and servers by simply running the desired number of instances of the single-camera processing container. We support a drop-in replacement for our detection and re-id feature extraction models to offer enough flexibility for consumers to evaluate different models on real-world data of the target domain.

Except for the data storage container, the Docker containers build on the same base image. The data storage container is based on the latest official MySQL server Docker image. For the processing Docker image, we use the base image *pytorch/pytorch:1.8.1-cuda10.2-cudnn7-devel* which supports running CNNs within a Docker container on the host system's GPU. It includes *Python 3.7.10* with *PyTorch 1.8.1* and is provided by PyTorch on Docker Hub. To build our image from the base image, we include application code as well as the trained CNN models. The containers are interconnected either locally via a Docker network or over via their IP addresses. To customize and set up our system, we provide configuration files. This allows for the seamless integration and removal of processed camera streams, by simply creating a configuration file and starting a single-camera processing container or stopping it.

6. Evaluation

We conduct our experiments on two systems with affordable components. **Setup 1** contains NVIDIA GTX 1080 Ti graphics cards along with 64GB RAM and the Intel Xeon E5-2630 CPU. **Setup 2** consists of the NVIDIA RTX 2080 Ti, 32GB RAM and the Intel Core i7-9700K.

Computation times are measured based on the first video of the test set of the MTA dataset [22]. If not stated otherwise, we use 320×320 pixels as input size for detection and the SORT+TMM tracker. Both CNN models operate in half-precision mode. The detector processes 30 frames per batch. The feature extractor processes the detections within these 30 frames at once as long as a maximum batch size of 64 is not exceeded.

6.1. Detection

For better generalization of the person detector towards surveillance scenarios, we fine-tune a YOLOv4 [5] model on the PANDA dataset [49], which is a gigapixel video dataset for real-world human-centric approaches and contains scenes with a large field of view and large crowds. These characteristics make it more suitable for transfer learning than other common recognition datasets such as COCO [27]. We choose YOLOv4, which is a one-stage person detector, due to its favorable speed against two-stage detectors and its superior accuracy against other one-stage detectors for person detection in surveillance context [9]. The dataset consists of a video subset of 15 video sequences at 2 FPS (10 for training, 5 for testing) and an image subset composed of 555 static images (390 for training and 165 for testing). Due to the organization of competitions, the annotations for both test sets are kept private. To prevent overfitting, we extract an image validation set with 134 images as described in [9]. For training, we scale each image with a factor of 0.25 and generate crops with a resolution of 1920×1080 px. The YOLOv4 model is trained with Darknet using the default training settings, an input size of 960×544 px, and 6,000 iterations.

(a) Trained on COCO [27] dataset.

(b) Trained on PANDA [49] dataset.

Figure 3: YOLOv4 detection results on real-world surveillance data for different training data. Figure 3b shows an improved detection recall for people far away from the camera and occlusions. However, it produces false positives for mirrored reflections.

Image Size	Prec.	Setup 1 (ms)	Setup 2 (ms)
160 × 160	FP16	$\mathbf{6 \pm 1}$	$\mathbf{5 \pm 0}$
320 × 320		9 ± 1	6 ± 0
640 × 640		19 ± 1	10 ± 1
160 × 160	FP32	$\mathbf{6 \pm 1}$	$\mathbf{5 \pm 0}$
320 × 320		10 ± 1	8 ± 0
640 × 640		24 ± 1	19 ± 1

Table 1: Detector processing frametimes largely depend on the the input image size. The table also shows that the performance gain from a half precision model is most prominent for our biggest image size.

Tracker	Setup 1 (ms)	Setup 2 (ms)
SORT	$\mathbf{1 \pm 1}$	$\mathbf{1 \pm 1}$
SORT+TMM	$\mathbf{1 \pm 1}$	$\mathbf{1 \pm 1}$
DeepSORT	15 ± 88	6 ± 5

Table 2: Tracker processing frametimes are not significantly influenced by our TMM.

Figure 3 provides a qualitative comparison of models trained on COCO and PANDA and applied to real-world surveillance data. Results indicate that both detectors report accurate results for people close to the camera. However, using the model trained with PANDA is beneficial for detecting small people in the background due to the datasets characteristics. It is worth noting that the COCO model does not generate any false positive detection, whereas the PANDA model produces a few, especially in the background (*e.g.* reflection in the window). We decide to use the PANDA model since a high detection recall assures that most persons in the scene are available for search. Finally, we observe no difference between full- (FP32) and half-precision (FP16) processing for detection accuracy.

Table 1 presents frame processing times for different parameter combinations. The input image size considerably influences the computation time. Since our model delivers promising results even with an input image resolution of 320 × 320 px, we are theoretically able to detect persons in more than 100 frames per second which is sufficient to meet the real-time requirement. The difference between FP32 and FP16 computation is small for low input resolutions and becomes more significant as the image size increases. However, since no difference in detection accuracy is observable, it is advantageous to use the faster FP16 computation in real-world application.

6.2. Single-camera Tracking

In this section, we compare the processing times of three different tracking variants. Besides SORT, we consider our enhanced version with the TMM and DeepSORT. Concerning tracking metrics, DeepSORT outperforms the SORT algorithm on the MOT benchmarks and produces fewer fragmentations. Track fragmentation is acceptable to a certain extent for person search systems. Fragmented tracks only lead to multiple results of the same person in the search ranking. Identity switches have worse effects since they corrupt the track features used for the retrieval. Results in Table 2 show that the computation time of using either the original or the improved version of SORT is negligible. Distance computations between visual features rarely need to be carried out in the TMM variant, so the overhead has minimal impact. In contrast, processing a frame with the DeepSORT algorithm took 15ms and 6ms on average and thus, influences the real-time capability.

The second row in Figure 4 depicts qualitative tracking results (SORT+TMM). Images are sampled evenly from the

Backbone	mAP	R-1
MobileNetv2	16.6 ± 8.0	31.3 ± 18.2
ResNet-18	15.9 ± 7.1	31.1 ± 17.4
ResNet-50	16.2 ± 7.1	30.5 ± 16.8
ResNet-101	17.4 ± 7.8	32.4 ± 18.2
OSNet-x0.25	18.8 ± 8.1	34.8 ± 18.8
OSNet-x0.5	21.4 ± 10.0	38.8 ± 21.0
OSNet-x0.75	22.6 ± 10.4	40.1 ± 21.6
OSNet-x1.0	$\mathbf{24.3 \pm 10.4}$	$\mathbf{42.3 \pm 20.8}$

Table 3: Generalization of person re-id models after training on two datasets and validating on a third. The OSNet-x1.0 performs favorably compared to other model architectures.

Method	mAP	R-1
Baseline	24.3 ± 10.4	42.3 ± 20.8
Random Dataset Sampler (RDS)	26.0 ± 11.7	45.2 ± 23.0
Color Jitter (CJ)	25.3 ± 12.7	44.9 ± 22.8
Random Cropping (RC)	24.2 ± 10.7	42.1 ± 20.9
Random Erasing (RE)	24.8 ± 9.4	43.1 ± 18.8
Random Patching (RP)	23.6 ± 9.0	41.7 ± 18.8
Random Rotation (RR)	25.5 ± 9.6	43.7 ± 19.8
RDS + CJ + RR + RE	26.4 ± 9.7	45.0 ± 18.4
RDS + CJ + RR (FP32)	$\mathbf{26.9 \pm 10.6}$	45.6 ± 20.8
RDS + CJ + RR (FP16)	$\mathbf{26.9 \pm 10.6}$	$\mathbf{45.7 \pm 20.8}$

Table 4: From the individual data augmentation and training strategies, random dataset sampling leads to the largest improvements for the leave-one-out cross-validation.

tracks. The shown tracks last for multiple seconds and do not break off or switch if the person walks through shaded areas or next to another person.

6.3. Re-identification

We use transfer learning for training our person re-id models. Therefore, we selected three research datasets with characteristics similar to real-world surveillance imagery and diversity regarding camera views. Concretely, we fine-tune our models on Market-1501 [62], DukeMTMC-reID [39], and PRAI-1581 [60]. Market-1501 and DukeMTMC represent flat camera views with mostly good resolution, whereas PRAI-1581 is a drone dataset captured from high altitudes. Hence, it provides low-resolution footage of persons from steep views. We perform leave-one-out cross-validation, *i.e.*, we use two datasets for training and the third one for testing in each run. Mean and standard deviations for the mean average precision (mAP) and the rank-1 accuracy (R-1) are reported as evaluation metrics.

We compare generalization abilities of different backbone models in Table 3. For training the models, we rely on the original setups and settings from the papers. Since multiple works showed that baseline methods are able to outperform complex state-of-the-art models [32, 56], especially concerning generalization on unseen domains [22], similar baseline approaches are used in our experiments. The OSNet [64] outperforms the MobileNetv2 [40] and the ResNet [15] models by a large margin, despite the smaller network size in comparison with *e.g.* ResNet-50 or ResNet-101. In total, OSNet-x1.0 achieves the best results.

To further improve this model, we evaluate different data augmentation methods and the random dataset sampler (RDS) in Table 4. The random dataset sampling procedure is beneficial compared to typical random sampling. With the RDS, each batch contains an equal number of images from the datasets and therefore regularizes the CNN with regarding different domains. Except for random cropping and random patching [64], all data augmentation strategies improve the re-id. Color jitter changes the brightness, contrast, and saturation of the images and thus leads to robustness against different lighting conditions. Randomly rotating images during the training of re-id models was proposed by Moritz et al. [35], especially to simulate different viewing angles for cameras in high altitudes. Surprisingly, random erasing leads to an improvement despite other works [32] reporting that it harms generalization by forcing the CNN to overfit data to cope with artificial occlusions. However, if the best working approaches are combined, random erasing deteriorates the results, and thus we do not use it for training the final model. Reducing the precision of the model does not lead to a significant change in performance but slightly lowers the computation time (see Table 4 and Table 5). Hence, the use of the FP16 model is advantageous.

Precision	Setup 1 (ms)	Setup 2 (ms)
FP16	$\mathbf{15 \pm 28}$	$\mathbf{10 \pm 16}$
PF32	17 ± 34	11 ± 14

Table 5: Feature extractor processing frametimes improve slightly with a half precision model.

We present a qualitative retrieval result of our person search system with real-world data in Figure 4. The image on the left depicts the query image, while the other ones display the first ten positions in the ranking. The real-world dataset consists of 30,000 tracks from three cameras. Multiple sequences per camera were used, so that the person-of-interest occurred nine times in total. The first ranking positions contain eight matches from all of the three cameras. The model generates visual features that are independent of the camera characteristics. The bottom row visualizes that each search result represents a track and not only a single image.

6.4. Real-time Capability

This section evaluates the pipeline speed measured in FPS, intending to assess the real-time capability. We com-

Figure 4: Person re-id results on real-world surveillance data. The first row shows a top 10 ranking for a query image, of tracks marked with a green (match) or red (mismatch) border. The bottom row depicts sample images from the SORT+TMM generated tracks.

Det. Size	Tracker	Setup 1 (FPS)	Setup 2 (FPS)
160 × 160	SORT+TMM	**49.0**	**76.4**
	DeepSORT	30.3	51.3
320 × 320	SORT+TMM	38.2	59.0
	DeepSORT	19.0	48.3
640 × 640	SORT+TMM	25.8	40.4
	DeepSORT	12.4	30.4

Table 6: The overall system performance on the MTA dataset [22] as described in Section 6. Even our more affordable setup 1 (GTX 1080 Ti) achieves real-time processing speeds with a reasonable detector size of 320 × 320 px and the SORT+TMM tracker.

pare the most important design choices concerning the trade-off between speed and accuracy, namely detector input size and the selection of the tracking component. Results are presented in Table 6. We argue that at least 35 FPS are required in our experiments to ensure real-time processing for real-world camera streams sending 30 FPS. The processing speed very much depends on the number of people visible in the scene. Therefore, time buffers enable handling situations when more persons than usual show up.

In general, the second hardware setup is significantly faster due to the improved computational power of the newer graphics card. The setup allows applying the DeepSORT tracker and forwarding images of size 320 × 320 px in real-time, even for crowded scenes. Moreover, a detector input size of 640 × 640 px is possible in conjunction with the SORT+TMM tracker, which improves the detection of persons in the background. Using this image size together with DeepSORT leads to a processing speed of 30.4 FPS, where crowded scenes possibly cause frames to be skipped. Our first hardware setup achieves real-time capability as well for a detector image size of 320 × 320 px images and the SORT+TMM tracker. In summary, our system meets real-time requirements with affordable hardware and can process camera streams with up to 76 FPS.

7. Conclusion

In this work, we have presented a real-world person search system for large-scale surveillance scenarios. We addressed specific challenges posed by surveillance footage in uncontrollable environments as well as further real-world requirements such as real-time processing and a high degree of flexibility. Using CNNs, specifically selected and trained concerning the trade-off between generalization and speed, paired with improved track association without computational overhead through our TMM, our system achieves real-time processing speeds with affordable hardware setups. The system processes camera streams with up to 76 FPS, while more balanced settings in terms of quality and processing speed still exceed 35 FPS without using the most recent hardware. Our system offers a blueprint with a set of techniques and best practices for developing a complete person search pipeline tailored to the domain generalization challenge of person re-identification without sacrificing cost-effective deployment options and affordable hardware requirements. The practicality of our system on unseen real-world data has become evident through our qualitative analysis with unlabeled real-world surveillance data.

Acknowledgment

This work was supported by the Mannheim Police Headquarters. Together with the state of Baden-Württemberg and the Mannheim Police Headquarters an intelligent vision-based activity recognition will be tested and further developed in a model project in Mannheim until 2023.

References

[1] P. Bergmann, T. Meinhardt, and L. Leal-Taixé. Tracking without bells and whistles. pages 941–951, 2019.

[2] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *2016 IEEE international conference on image processing (ICIP)*, pages 3464–3468. IEEE, 2016.

[3] Erik Bochinski, Volker Eiselein, and Thomas Sikora. High-speed tracking-by-detection without using image information. In *International Workshop on Traffic and Street Surveillance for Safety and Security at IEEE AVSS 2017*, Lecce, Italy, Aug. 2017.

[4] Erik Bochinski, Tobias Senst, and Thomas Sikora. Extending iou based multi-object tracking by visual information. In *IEEE International Conference on Advanced Video and Signals-based Surveillance*, pages 441–446, Auckland, New Zealand, Nov. 2018.

[5] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[6] Peng Chen, Jing Liu, Bohan Zhuang, Mingkui Tan, and Chunhua Shen. Aqd: Towards accurate quantized object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 104–113, June 2021.

[7] Seokeon Choi, Taekyung Kim, Minki Jeong, Hyoungseob Park, and Changick Kim. Meta batch-instance normalization for generalizable person re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3425–3435, 2021.

[8] Mickael Cormier, Dmitrii Seletkov, and Jürgen Beyerer. Towards lower precision quantization for pedestrian detection in crowded scenario. In *IEEE EUROCON 2021-19th International Conference on Smart Technologies*, pages 254–258. IEEE, 2021.

[9] Mickael Cormier, Stefan Wolf, Lars Sommer, Arne Schumann, and Jürgen Beyerer. Fast pedestrian detection for real-world crowded scenarios on embedded gpu. In *IEEE EUROCON 2021 - 19th International Conference on Smart Technologies*, 2021.

[10] Patrick Dendorfer, Hamid Rezatofighi, Anton Milan, Javen Shi, Daniel Cremers, Ian Reid, Stefan Roth, Konrad Schindler, and Laura Leal-Taixé. Mot20: A benchmark for multi object tracking in crowded scenes. *arXiv preprint arXiv:2003.09003*, 2020.

[11] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6569–6578, 2019.

[12] Matteo Fabbri, Guillem Braso, Gianluca Maugeri, Orcun Cetintas, Riccardo Gasparini, Aljosa Osep, Simone Calderara, Laura Leal-Taixe, and Rita Cucchiara. Motsynth: How can synthetic data help pedestrian detection and tracking? In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021.

[13] Matteo Fabbri, Fabio Lanzi, Simone Calderara, Andrea Palazzi, Roberto Vezzani, and Rita Cucchiara. Learning to detect and track visible and occluded body joints in a virtual world. In *European Conference on Computer Vision (ECCV)*, 2018.

[14] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[16] Xin Huang, Zheng Ge, Zequn Jie, and Osamu Yoshie. Nms by representative region: Towards crowded pedestrian detection by proposal pairing. In *CVPR*, 2020.

[17] Khawar Islam. Person search: New paradigm of person re-identification: A survey and outlook of recent works. *Image and Vision Computing*, 101:103970, 2020.

[18] Boseung Jeong, Jicheol Park, and Suha Kwak. Asmr: Learning attribute-based person search with adaptive semantic margin regularizer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12016–12025, 2021.

[19] Jieru Jia, Qiuqi Ruan, and Timothy M Hospedales. Frustratingly easy person re-identification: Generalizing person re-id in practice. *arXiv preprint arXiv:1905.03422*, 2019.

[20] Xin Jin, Cuiling Lan, Wenjun Zeng, Zhibo Chen, and Li Zhang. Style normalization and restitution for generalizable person re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3143–3152, 2020.

[21] Jung Uk Kim, Sungjune Park, and Yong Man Ro. Robust small-scale pedestrian detection with cued recall via memory learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3050–3059, October 2021.

[22] Philipp Kohl, Andreas Specker, Arne Schumann, and Jurgen Beyerer. The mta dataset for multi-target multi-camera pedestrian tracking by weighted distance aggregation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 1042–1043, 2020.

[23] SV Aruna Kumar, Ehsan Yaghoubi, Abhijit Das, BS Harish, and Hugo Proença. The p-destre: A fully annotated dataset for pedestrian detection, tracking, and short/long-term re-identification from aerial devices. *IEEE Transactions on Information Forensics and Security*, 16:1696–1708, 2020.

[24] Laura Leal-Taixé, Anton Milan, Ian Reid, Stefan Roth, and Konrad Schindler. Motchallenge 2015: Towards a benchmark for multi-target tracking. *arXiv preprint arXiv:1504.01942*, 2015.

[25] Wei Li, Rui Zhao, Tong Xiao, and Xiaogang Wang. Deepreid: Deep filter pairing neural network for person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 152–159, 2014.

[26] Shengcai Liao and Ling Shao. Interpretable and generalizable person re-identification with query-adaptive convolution

and temporal lifting. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 456–474. Springer, 2020.

[27] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[28] Weiyao Lin, Huabin Liu, Shizhan Liu, Yuxi Li, Rui Qian, Tao Wang, Ning Xu, Hongkai Xiong, Guo-Jun Qi, and Nicu Sebe. Human in events: A large-scale benchmark for human-centric video analysis in complex events. *arXiv preprint arXiv:2005.04490*, 2020.

[29] Songtao Liu, Di Huang, and Yunhong Wang. Adaptive nms: Refining pedestrian detection in a crowd. In *CVPR*, 2019.

[30] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, et al. Ssd: Single shot multibox detector. In *ECCV*, 2016.

[31] Ruiqi Lu, Huimin Ma, and Yu Wang. Semantic head enhanced pedestrian detection in a crowd. *Neurocomputing*, 2020.

[32] Hao Luo, Youzhi Gu, Xingyu Liao, Shenqi Lai, and Wei Jiang. Bag of tricks and a strong baseline for deep person re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[33] Jiayuan Mao, Tete Xiao, Yuning Jiang, and Zhimin Cao. What can help pedestrian detection? In *CVPR*, 2017.

[34] Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. Mot16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*, 2016.

[35] Lennart Moritz, Andreas Specker, and Arne Schumann. A study of person re-identification design characteristics for aerial data. In *Pattern Recognition and Tracking XXXII*, volume 11735, page 117350P. International Society for Optics and Photonics, 2021.

[36] B. Pang, Y. Li, Y. Zhang, M. Li, and C. Lu. Tubetk: Adopting tubes to track multi-object in a one-step training model. pages 6308–6318, 2020.

[37] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[38] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015.

[39] Ergys Ristani, Francesco Solera, Roger S. Zou, R. Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *ECCV Workshops*, 2016.

[40] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[41] Adam Satariano. London police are taking surveillance to a whole new level. `https://www.nytimes.com/2020/01/24/business/london-police-facial-recognition.html`, *Accessed: October 11, 2021*.

[42] Shuai Shao, Zijian Zhao, Boxun Li, Tete Xiao, Gang Yu, Xiangyu Zhang, and Jian Sun. Crowdhuman: A benchmark for detecting human in a crowd, 2018.

[43] Amarjot Singh, Devendra Patil, and SN Omkar. Eye in the sky: Real-time drone surveillance system (dss) for violent individuals identification using scatternet hybrid deep learning network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1629–1637, 2018.

[44] Jifei Song, Yongxin Yang, Yi-Zhe Song, Tao Xiang, and Timothy M Hospedales. Generalizable person re-identification by domain-invariant mapping network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 719–728, 2019.

[45] Vladislav Sovrasov and Dmitry Sidnev. Building computationally efficient and well-generalizing person re-identification models with metric learning. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 639–646. IEEE, 2021.

[46] Andreas Specker and Jürgen Beyerer. Improving attribute-based person retrieval by using a calibrated, weighted, and distribution-based distance metric. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 2378–2382. IEEE, 2021.

[47] Lucas Steinmann, Lars Sommer, Arne Schumann, and Jürgen Beyerer. Fast and lightweight person detector for unmanned aerial vehicles.

[48] Masato Tamura and Tomokazu Murakami. Augmented hard example mining for generalizable person re-identification. *arXiv preprint arXiv:1910.05280*, 2019.

[49] Xueyang Wang, Xiya Zhang, Yinheng Zhu, Yuchen Guo, Xiaoyun Yuan, et al. Panda: A gigapixel-level human-centric video dataset. In *CVPR*, 2020.

[50] Zhongdao Wang, Liang Zheng, Yixuan Liu, and Shengjin Wang. Towards real-time multi-object tracking. *arXiv preprint arXiv:1909.12605*, 2019.

[51] Longhui Wei, Shiliang Zhang, Wen Gao, and Qi Tian. Person transfer gan to bridge domain gap for person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 79–88, 2018.

[52] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3645–3649. IEEE, 2017.

[53] Alexander Womg, Mohammad Javad Shafiee, Francis Li, and Brendan Chwyl. Tiny ssd: A tiny single-shot detection deep convolutional neural network for real-time embedded object detection. In *CRV*, 2018.

[54] Zhujun Xu, Emir Hrustic, and Damien Vivet. Centernet heatmap propagation for real-time video object detection. In *European Conference on Computer Vision*, pages 220–234. Springer, 2020.

[55] Mang Ye, Jianbing Shen, Gaojie Lin, Tao Xiang, Ling Shao, and Steven C. H. Hoi. Deep learning for person re-identification: A survey and outlook.

[56] Mang Ye, Jianbing Shen, Gaojie Lin, Tao Xiang, Ling Shao, and Steven C. H. Hoi. Deep learning for person re-

identification: A survey and outlook. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[57] Liliang Zhang, Liang Lin, Xiaodan Liang, and Kaiming He. Is faster r-cnn doing well for pedestrian detection? In *ECCV*, 2016.

[58] Shanshan Zhang, Rodrigo Benenson, and Bernt Schiele. Citypersons: A diverse dataset for pedestrian detection, 2017.

[59] Shifeng Zhang, Longyin Wen, Xiao Bian, Zhen Lei, and Stan Z Li. Occlusion-aware r-cnn: detecting pedestrians in a crowd. In *ECCV*, 2018.

[60] Shizhou Zhang, Qi Zhang, Yifei Yang, Xing Wei, Peng Wang, Bingliang Jiao, and Yanning Zhang. Person re-identification in aerial imagery. *IEEE Transactions on Multimedia*, 23:281–291, 2020.

[61] Yuyang Zhao, Zhun Zhong, Fengxiang Yang, Zhiming Luo, Yaojin Lin, Shaozi Li, and Nicu Sebe. Learning to generalize unseen domains via memory-based multi-source meta-learning for person re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6277–6286, 2021.

[62] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *Proceedings of the IEEE international conference on computer vision*, pages 1116–1124, 2015.

[63] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13001–13008, 2020.

[64] Kaiyang Zhou, Yongxin Yang, Andrea Cavallaro, and Tao Xiang. Omni-scale feature learning for person re-identification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3702–3712, 2019.

[65] Kaiyang Zhou, Yongxin Yang, Andrea Cavallaro, and Tao Xiang. Learning generalisable omni-scale representations for person re-identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[66] X. Zhou, V. Koltun, and P. Krähenbühl. Tracking objects as points. pages 474–490, 2020.

[67] Jinguo Zhu, Zejian Yuan, Chong Zhang, Wanchao Chi, Yonggen Ling, et al. Crowded human detection via an anchor-pair network. In *WACV*, 2020.