

PAF-NET: AN END-TO-END PEOPLE ANALYTICS FRAMEWORK INTEGRATING MULTI-MODAL BIOMETRIC IDENTIFICATION FOR CLOSED-AREA VIDEO SURVEILLANCE

MARWA EL-NASHAR^{1,2}, MOHAMED ROHAIM¹, KHAIRY ASSAR¹, ALY M. EL-SEMARY¹

¹ Systems and Computer Engineering Department, Faculty of Engineering, Al-Azhar University, Cairo, Egypt

² Computer Engineering Department, Faculty of Engineering, Cairo University, Giza, Egypt

E-mail: ¹marwa.el-nashar82@eng1.cu.edu.eg, mrohaim@azhar.edu.eg, khairyassar@azhar.edu.eg, aelsemary@azhar.edu.eg

ABSTRACT

Automated people's identification in institutional environments is challenging when facial information is unavailable or unreliable. This paper presents the People Analytics Framework (PAF-Net), a multi-camera system that combines person detection and tracking, face detection and verification, and person re-identification. The system uses facial evidence when a usable face is available. When facial information is unavailable, the system uses person re-identification to identify the individual based on appearance. The framework was evaluated using a locally collected campus dataset from five recording sessions and four cameras. The dataset contains 671 person tracks and 24,721 person-crop images. In total, 1,695 identification queries were evaluated. A usable face was available in 287 queries, while the remaining 1,408 queries were processed using person Re-ID. The system correctly identified 280 face-based queries and 1,393 Re-ID queries, giving an overall identification accuracy of 98.7%.

Keywords: *Face Detection, Face Verification, Person Re-Identification, Surveillance System, Deep Learning*

1. INTRODUCTION

Computer Vision (CV) is an important transformative technology that enables smart machines to interact with their surroundings by understanding what they 'see' through detection and verification algorithms. Recently, CV has been extensively researched across a wide spectrum of fields, including public services, consumer electronics, industrial automation, medical imaging, and national security, underscoring the broad societal importance of reliable, scalable visual intelligence. and smart video surveillance. Smart Video Surveillance systems (SVSs) are among the most important areas of CV. They aim to identify and report various cases of digital image and video detection and recognition using traditional and deep learning methods [1].

SVS is a famous application for CV. It is used for monitoring in public and private places. Nowadays, it is used to detect, recognize, and track objects, humans, and cars. etc., in homes, offices, hospitals, malls, and parking areas, depending on the preference of the user [2]. Outdoor and indoor are

the two main environmental types of SVSs. Outdoor surveillance mainly involves detecting distant objects and is designed to handle long-range detection under challenging atmospheric conditions. On the other hand, indoor surveillance focuses on detecting, identifying, counting, clustering, and recognizing people, as well as their behaviors and actions [2].

Automated Human Surveillance Systems (AHSS) are among the most well-known SVS systems. They rely on observing and monitoring humans offline or in real time without requiring constant human oversight. They can detect, track, and verify unusual or suspicious behavior and raise alarms without human monitoring. Recently, the state of the art in AHSS uses more advanced methods such as convolutional object detection [3, 4], multi-object tracking algorithms [5, 6, 7], anomaly detection frameworks [8], facial analysis pipelines [9, 10, 11, 12, 13], and person re-identification models [14, 15]. Most existing systems rely heavily on facial visibility to establish identity. However, facial recognition fails with obscured faces and poor-quality images caused by blur, low resolution, poor

lighting, or adverse conditions [16]. A robust integrated pipeline is therefore required.

This study addresses the challenge of identifying the same individual across multiple non-overlapping cameras when facial information is unavailable or insufficient for reliable recognition. In surveillance environments, an individual may be clearly visible in one camera or frame, while appearing in another view under conditions where the face is occluded, poorly captured, or otherwise unsuitable for facial identification.

Face-based systems therefore cannot identify every observation, while appearance-based person re-identification (Re-ID) systems may overlook the additional identity information provided when a face is visible. A more practical challenge is that person detection, tracking, face analysis, and Re-ID are often studied and evaluated as separate tasks using public benchmark datasets. This leaves a gap between individual model performance, and the practical question of how these different components should be selected, combined, and used together in a real multi-camera surveillance environment.

To address this problem, this paper introduces the People Analytics Framework (PAF-Net), an identity-centric machine-learning pipeline designed for one or more video cameras operating within a controlled private network. PAF-Net integrates person detection, multi-camera tracking, face detection and clustering, face-based identity verification, and full-body appearance-based person re-identification within a unified workflow. The framework is designed to associate observations across cameras and over time and to determine the most appropriate identification pathway based on the available visual evidence. In this way, the system is intended to support three fundamental questions for detected individuals: Who are they? Where were they seen? When were they there?

The main objectives are to develop an end-to-end system that brings together person detection and tracking, face detection, face clustering and verification, and full-body person re-identification. In addition, it aims to establish a routing process that determines, for each tracked person, whether face-based verification can be used or whether appearance-based re-identification is needed when facial information is unavailable. Moreover, the study compares candidate models for each module under the same conditions, using data collected in a real multi-camera campus environment, to select the most suitable model for each stage. Finally, it evaluates the selected models and the complete

system in terms of both identification accuracy and processing time.

This work is important in three main respects. The first is that it addresses a practical limitation of face-based surveillance by providing an alternative way of identifying individuals when reliable facial information is unavailable. This approach helps preserve identity information across a range of observations, including cases where a person's face alone is insufficient for facial identification. Second, the study presents a systematic approach to model selection for the main stages of the proposed pipeline.

The evaluation considers four detector-tracker configurations, four face-detection models, nine face-verification models, and eleven person-re-identification configurations. Testing these alternatives under consistent real conditions enables direct performance comparisons and helps determine which components work together correctly within the proposed system. Third, the framework is proposed for closed, private-network environments, including institutional campuses, hospitals, and schools. In these settings, video and identity data remain within the organization's infrastructure, helping meet the operational and privacy requirements of institutional surveillance [17].

The main contributions of this study are: (1) an identity-centric multi-camera integration framework combining face-visible and face-invisible identification; (2) a systematic procedure for selecting and integrating detection, tracking, face, and Re-ID components under a common institutional scenario; (3) an identity-association mechanism that combines available visual evidence with temporal and camera information; and (4) an experimental protocol that evaluates both individual components and the complete integrated framework.

This paper is organized as follows: Section 2 provides a literature review, Section 3 introduces the proposed framework, and Section 4 presents the results.

2. RELATED WORK

Deep learning (DL) is now widely used in video surveillance and remote person recognition. CNNs are used commonly to extract visual features from images and videos. These features can then be used for person identification and other surveillance applications.

2.1 Surveillance Systems and Frameworks

Elharrouss et al. [18] reviewed the architecture of video surveillance systems and examined their use in motion detection, scene understanding, and behavior analysis. Their review showed that different surveillance tasks require different methods. This supports the use of a modular pipeline, as adopted in the present work.

Zahrawi and Shaalan [19] studied automatic weapon detection in indoor environments, including banks, malls, and supermarkets. They compared several object detection methods, including YOLO and Single Shot MultiBox Detector (SSD). Their results showed that modern object detectors can be used for real-time threat detection when they are properly adapted to the target environment.

Ardabili et al. [17] developed a surveillance system using YOLOv5 for person detection, ByteTrack for tracking, and graph convolutional networks for action recognition. The authors also considered privacy and ethical issues in automated surveillance and highlighted the importance of protecting personal data. These considerations are relevant to PAF-Net because the proposed system is designed for a closed and private environment.

He et al. [20] introduced Mask R-CNN, a two-stage detection method with an additional branch for generating pixel-level object masks. Its ability to produce instance-level masks has made it useful in different monitoring applications. However, its two-stage structure generally makes it slower than single-stage detectors. Its reported results were also obtained on the COCO benchmark, which does not fully represent the conditions found in real multi-camera surveillance environments.

The studies reviewed above mainly focus on object detection, threat detection, and activity recognition. In many cases, person identification is treated as a separate task and relies on facial information. This creates a limitation when the person's face is not visible or cannot be detected clearly.

2.2 Person Detection and Multi-Object Tracking

Angadi et al. [21] examined traditional methods for object detection and tracking, such as background subtraction, optical flow, and spatial filtering. These methods are simple and use limited computational

resources, but they can be affected by occlusion, lighting, and background movement.

DL methods have become widely used for person detection in surveillance systems. CNN-based detectors provide the speed needed for processing video frames, and the YOLO family is commonly used for this purpose [19], [17]. Detection alone, however, cannot maintain a person's identity across consecutive frames. Tracking methods are therefore needed to connect detections over time [5].

Tracking-by-detection methods such as DeepSort [6] combine motion information with appearance features to associate detections between frames. ByteTrack also uses low-confidence detections when matching objects to existing tracks, which can help maintain tracks when people are partially occluded or appear blurred. Previous comparisons have shown that tracker selection can affect the quality of the resulting tracks [7].

Most tracking studies focus on tracking performance rather than its effect on later identification. In an identity-based system, tracking errors can also affect the final identity result. For example, an identity switch can cause observations from two different people to be stored under the same track.

2.3 Face Detection and Recognition

Face detection and recognition methods have also developed considerably with the use of deep learning. Cascade-based detectors are relatively fast but can be affected by changes in pose and illumination. MTCNN uses a multi-stage convolutional approach to detect faces and facial landmarks. Later extensions have been proposed to improve detection under more difficult conditions [9].

For face recognition, FaceNet [11] learns an embedding representation using a triplet-based objective. Face verification can then be performed by comparing the distance between two face embeddings. ArcFace [10] introduces an additive angular margin to improve the separation between different identities. Reviews of face recognition methods [12], [13] have also pointed out that results obtained on controlled benchmarks may not reflect performance in surveillance environments.

Singh et al. [22] investigated real-time facial recognition in surveillance applications by fine-tuning VGGFace on a dataset containing 7,500 images from 26 subjects. They reported a recognition accuracy of 96%. Their work showed that transfer learning can be useful when a general-purpose face recognition model is adapted to a specific environment.

Despite improvements in face recognition, the method has an important limitation. If a person's face is not visible or the captured face is not clear enough, face recognition cannot provide a reliable identity result.

2.4 Person Re-Identification

Nalty et al. [16] reviewed dl methods for remote person recognition, including face verification, person re-identification, and gait analysis. They noted that long-distance imaging is affected by factors such as atmospheric conditions, motion blur, low resolution, pose, and lighting.

Person re-identification (Re-ID) provides another way to identify a person by comparing their overall appearance across different cameras. Unlike face recognition, Re-ID does not require a clear view of the person's face. OSNet [14] is a commonly used Re-ID model that learns features at different scales and combines local and global appearance information. Its relatively lightweight design also makes it suitable as a Re-ID baseline.

Re-ID performance can degrade when the training and deployment environments differ. Changes in camera position, viewpoint, lighting, or clothing can affect the extracted appearance features [15]. In addition, many Re-ID studies evaluate the model using pre-cropped person images. This does not fully reflect a complete surveillance workflow, in which detection, tracking, and identity assignment occur before the Re-ID model is used.

2.5 Summary and Research Gap

Table 1 presents the main approaches discussed in this section, along with their limitations and their relevance to the proposed framework. Previous studies have developed effective methods for detection, tracking, face recognition, and person Re-ID. However, these methods are often studied as separate tasks. Fewer studies examine how these

components can be combined into a single identification system, especially when the person's face is not visible. PAF-Net combines these components in a single multi-camera pipeline for person identification.

Table 1: Summary Of Related Work And Limitations

Work	Approach	Limitation	How PAF-Net addresses it
Singh et al. [22]	Fine-tuned VGGFace for real-time face recognition (96%)	Identification depends on a visible face	Adds a full-body ReID pathway for face-absent folders
Zahrawi & Shaalan [19]	YOLO / SSD weapon detection in indoor surveillance	Detects threats, not the identity of persons	Links every detection to a person record (who, where, when)
Elharrouss et al. [18]	Review of surveillance architectures	No single architecture excels across all tasks	Modular pipeline with evidence-based model selection per stage
Angadi et al. [21]	Classical detection and tracking (background subtraction, optical flow)	Sensitive to occlusion, illumination, and dynamic backgrounds	Deep detector-tracker (YOLOv5 + ByteTrack)
Ardabili et al. [17]	Cloud SVS: YOLOv5, ByteTrack, GCN action recognition	Focus on actions; raises privacy and ethics concerns	Closed, private-network deployment with identity-level analysis
He et al. [20]	Mask R-CNN: two-stage detection with instance masks	Slower two-stage design; evaluated on a curated benchmark (COCO)	Single-stage detector, evaluated on real footage from four campus cameras
ReID studies [14], [15]	Appearance-based matching across cameras	Evaluated in isolation; sensitive to domain shift; ignores face evidence	ReID combined with face routing and selected on-campus data

Overall, previous studies show that person identification can be difficult when the face is not visible because of occlusion, pose, or camera angle. PAF-Net addresses this issue by combining face-based identification with appearance-based person Re-ID within a single system.

3. PROPOSED PEOPLE ANALYTICS FRAMEWORK (PAF-NET)

The People Analytics Framework (PAF-Net) is an end-to-end surveillance pipeline system designed for closed and private network environments, such as universities, hospitals, and schools. It collects videos from one or more cameras and processes the recordings using several deep learning modules. The system also stores information about detected individuals, enabling later queries about their identities, locations, and movement histories.

The PAF-Net pipeline involves four primary stages: (1) person detection and tracking, (2) face detection, (3) face clustering and verification, and (4) person re-identification. An integration layer combines the outputs of these stages into a unified person database. Figure 1 illustrates the overall system architecture.

The study uses an applied experimental research design consisting of four stages. First, the requirements of the target environment were defined, including multi-camera coverage, identification with or without visible faces, and private-network operation. Second, the system was divided into separate modules with defined inputs and outputs, allowing each module to be evaluated independently. Third, candidate models for each module were tested under the same conditions using data collected at the deployment site. The model providing the best balance between accuracy and processing time was selected. Finally, the selected models were integrated, and the complete system was evaluated. The evaluation procedure is described in Section 3.6.

3.1 Detection And Tracking Module (DTM)

The system takes videos from four cameras. Each frame is resized to $1,000 \times 600$ pixels before processing, and the frame rate is 30 fps. The detection and tracking module processes each frame to detect people, crop their images, and store the crops in separate folders. Each folder is linked to a unique tracker ID that records the person's path, frame number, and capture time. The records are updated as new frames are processed. Each person is represented by at least one image and may have images from different cameras and times.

However, not every captured image is usable for identification. Some crops cut off too much of the person, and others catch more than one person in the frame. To keep the data clean, we drop any image with a height-to-width ratio under 2:1 or with

multiple people in it. What's left is one person per image, and those are what we use for identification.

A query-based grouping process is applied to the cleaned database. A representative image is used to find and classify images of the same person under a common person ID. This process is repeated for all tracking folders. The resulting person table is used for identity queries, movement analysis, and system visualization.

To determine a suitable detection and tracking combination, four configurations were tested: YOLOv5 with DeepSort, YOLOv5 with ByteTrack, YOLOv8 with DeepSort, and YOLOv8 with ByteTrack. The configurations were compared based-on detection accuracy and processing speed for real-time multi-camera surveillance.

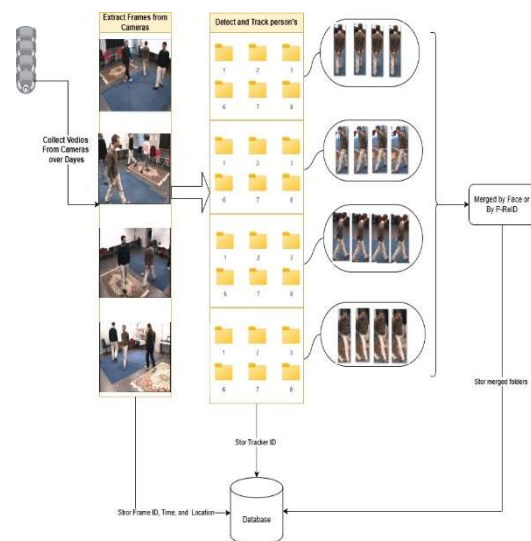


Figure 1: Architecture Of The People Analytics Framework Modules

3.2 Face Detection And Recognition Module (FDRM)

3.2.1 Face detection

Accurate face detection is a prerequisite for any identity verification step, and its quality directly shapes the reliability of everything that follows. To handle the range of image conditions encountered in a real surveillance environment, varying lighting, resolution, distance, and partial occlusion, we draw on the DeepFace library, which bundles several well-established face detection architectures under a unified interface. This study evaluates the OpenCV-based Haar Cascade classifier, the Single Shot MultiBox Detector (SSD), the Multi-task Cascaded Convolutional Neural Network (MTCNN), and RetinaFace. Dlib was also included in an initial

screening, as described in Section 4.2.1. Each model was pre-trained on large-scale face datasets, giving it a strong prior for detecting facial regions across diverse conditions.

The selection of a face detector for the final pipeline was not made on a single criterion. Rather, we evaluated each model across multiple test scenarios that reflect the kinds of inputs the system is likely to encounter in practice: folders containing only face images, mixed folders with both face and non-face images of the same person, and folders with no faces at all. This evaluation strategy is important because, within the PAF-Net pipeline, the face detection module does more than locate faces; it also serves as a routing gate, classifying each tracking folder as face-containing or face-absent and directing the latter group directly to the person re-identification module.

A detector that produces too many false positives would incorrectly route non-face images into the verification pipeline; one that misses too many faces would unnecessarily bypass verification. The chosen model must therefore balance recall, precision, and processing speed in roughly equal measures.

3.2.2 Face clustering

After face detection, images of the same person are grouped using unsupervised face clustering. Since identity labels are not available, the system uses visual similarity to group the images.

The model uses triplet loss and cluster loss to learn the embedding space. Triplet loss uses an anchor, a positive image from the same person, and a negative image from a different person, and pulls the anchor closer to the positive than to the negative.

$$L_{\text{triplet}} = \max(0, D(A, P) - D(A, N) + \alpha) \quad (\text{Equation 1})$$

Where A is the anchor image, P is a positive sample from the same identity, N is a negative sample from a different identity, D denotes the Euclidean distance between two different feature vectors, and α is a margin hyperparameter that enforces a minimum gap between positive and negative distances. By tuning α , we control how tightly the model must separate identities in the embedding space.

The cluster loss in Equation 2 reduces the differences between embeddings within the same cluster. It considers pairs of face embeddings and encourages them to have high cosine similarity.

$$L_{\text{cluster}} = \sum_i \sum_j (1 - \cos(f_i, f_j)) \quad (\text{Equation 2})$$

Where f_i and f_j are the two face feature vectors from the same cluster, and $\cos(\cdot)$ denotes cosine similarity. Minimizing this loss makes features from the same identity more similar. The two losses work together to separate distinct identities while keeping images of the same person close together.

The clustering stage uses VGG16, FaceNet, DeepID, SFace, ArcFace, and combinations of these models. These models differ in accuracy and computational cost.

3.2.3 Face verification

After clustering, the verification module checks whether a query face matches a known person in the database. This is a one-to-one comparison between two face images to determine whether they belong to the same person.

For this module, we adopt the ArcFace loss function to train the face embedding network. ArcFace modifies the standard softmax loss by introducing an additive angular margin m directly into the angle between the feature vector and the target class center. This geometric constraint tightens the distribution of same-class embeddings around their class centers. It widens the angular separation between different classes, producing embeddings that are substantially more discriminative than those learned under a conventional softmax objective. The ArcFace loss is formulated and defined in Equation 3.

$$L = -\frac{1}{N} \sum_{i=1}^N \log \left(\frac{e^{s(\cos \theta_{yi} + m)}}{e^{s(\cos \theta_{yi} + m)} + \sum_{i=1, i \neq y_i}^N e^{s(\cos \theta_j)}} \right) \quad (\text{Equation 3})$$

Where θ_{yi} is the angle between the feature vector of sample $*i*$ and the weight vector of its correct class, $*m*$ is the additive angular margin that enforces inter-class separation, and $*s*$ is a scale factor controlling the magnitude of the logits. θ_j denotes the angles associated with all non-target classes. The margin $*m*$ is the central design parameter of ArcFace: increasing it forces the model to learn more discriminative features, at the cost of slower convergence. In practice, the value of $*m*$ is chosen empirically based on the size and diversity of the training set.

The verification models VGG-Face, FaceNet, FaceNet512, OpenFace, DeepFace, DeepID, ArcFace, SFace, and GhostFaceNet are tested with Euclidean(L2), Euclidean, and Cosine similarity. Each model is evaluated using 100 positive and 100

negative pairs based on accuracy and average processing time.

3.3 Person Re-Identification Module (PRM)

Face-based identification cannot be used when a person's face is not visible because of camera angle, occlusion, or deliberate avoidance. To handle this case, PAF-Net includes a Person Re-Identification (ReID) module that uses full-body appearance, including clothing, body shape, gait, and posture. The module matches images of the same person across non-overlapping cameras captured at different times.

Besides the identification method, the person ReID in the PAF-Net system is used to clean data produced by the DMT process by deleting duplicate tracker IDs and split detections before identification

Person ReID can be applied to images or video sequences. Video-based ReID can provide more information about appearance and movement than a single image. Both approaches aim to learn feature representations that bring images of the same person closer together while separating images of different people, despite changes in camera, lighting, and pose.

We use a Deep Metric Learning (DML) framework to train the embedding network based on distances between feature representations. Two loss functions are considered: triplet loss, shown in Equation 1, and contrastive loss, shown in Equation 4.

The contrastive loss operates on pairs of images rather than triplets. For each pair, a Binary label indicates whether the two images share the same identity. or not, and the loss penalizes the model differently depending on that label:

$$L_{\text{Contrastive}} = \frac{1}{2N} \sum_{i=1}^N (y_i \cdot d_i^2 + (1 - y_i) \cdot \max(0, m - d_i)^2) \quad (\text{Equation 4})$$

Where $*N*$ is the total number of pairs in the batch, $y_i \in \{0, 1\}$ is the similarity label for pair $*i*$ (1 for same identity, 0 for different), d_i is the Euclidean distance between the two embeddings, and $*m*$ is a margin parameter that defines the minimum separation required between dissimilar pairs.

The two terms in the loss function deal with different types of pairs. The first term, $y_i d_i^2$, is used for similar pairs and reduces the distance between their embeddings. The second term, $(1 -$

$y_i) \max(0, m - d_i)^2$, is used for different pairs and penalizes distances that are smaller than the margin m . In this way, the loss brings images of the same person closer and keeps images of different people apart.

OSNet (Omni-Scale Network) is used as the backbone for person ReID. It learns features at different spatial scales, including details such as clothing patterns and larger features such as body shape. This is useful for multi-camera surveillance because the same person may appear at different sizes and resolutions. OSNet is trained using triplet loss, which gave better results than contrastive loss in our preliminary experiments. It also follows the same triplet-based approach used in the face module.

Several ReID models are tested to select the best model for PAF-Net. These included ResNet50, ResNet50-IBN, PCB-P6, SqueezeNet, DenseNet161, ShuffleNet, MobileNetV2, and several OSNet variants. The models are evaluated using Rank-1, Rank-5, Rank-10, Rank-20, and mean Average Precision (mAP) on data collected from the campus camera network.

3.4 System Integration

The research objective is to develop an end-to-end surveillance system, so integration of all phases is the most important part. Therefore, after collecting information from multiple cameras or at different times, perform face clustering or person re-identification in the query to cluster and merge all data under a person tracker ID, as shown in PAF-Net Algorithm 1.

Face detection, recognition, or person re-identification will be used mainly for users' requests. When a user provides an image of a person and asks to find or track them, one of the modules will determine which folder in the database contains an image of that person. After finding that person's folder, the system can answer any question the user asks: provide a trajectory of that person's time on campus or take that person's name from the user and add it to the folder if needed. Any time that happens, it saves verification data to a Verified Table in the database.

Finally, integration starts with selecting the best modules; in each phase of the system pipeline, different models are tested, and the model with the best performance is selected in the final system, then storing important information about each phase in a database as shown in Figure 2, the integration of the system of the analytics modules of people.

PAF-Net integrates the selected modules through a hierarchical identity-association procedure. For each person track, the system first determines whether sufficiently reliable facial evidence is available. When it is available, face verification provides the primary identity evidence. When facial evidence is unavailable or below the predefined confidence threshold, the system uses the person's appearance embedding for Re-ID. The resulting identity hypothesis is then checked against the configured temporal and camera constraints before an identity observation is stored. Each accepted observation contains the identity identifier, camera identifier, timestamp, track identifier, and confidence information. This integration layer defines how the independently developed vision models operate as one PAF-Net workflow.

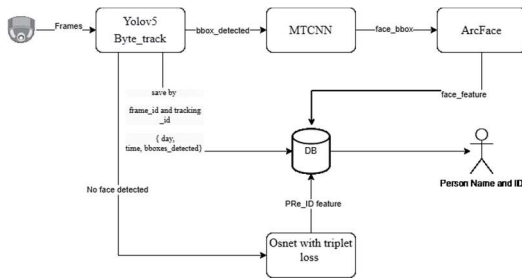


Figure 2: System Integration Of People Analytics Framework Modules

3.5 PAF-Net Algorithms

Algorithm 1: PAF-Net Processing Pipeline

1. Collect video streams from all deployed cameras.
2. Sample frames at 30 fps; resize to 1,000 × 600 pixels.
3. Apply ByteTrack-YOLOv5 for person detection and tracking.
4. Clean tracking folders:
 - a. Delete images with an aspect ratio < 2:1.
 - b. Delete images containing more than one person.
5. Store each person's images in the database with timestamps and location metadata.
6. For all camera folders:
 - a. If two folders contain images of the same person, merge their records.
 - b. Otherwise, create a new record in the database.
7. For each merged folder, apply MTCNN face detection:
 - a. If a face is detected, verify identity with SFace (Euclidean distance) against the database.
 - b. If no face is detected, or no confident match is returned, match the folder with OSNet (triplet loss).
8. Update the person record with the matched identity, camera locations, and timestamps; write confirmed identities to the Verified Table.
9. For a user query image, apply steps 7–8 and return the matched person's identity, trajectory, locations, and times.

3.6 Experimental Setup And Evaluation Protocol

Data collection. All experiments used video recorded by four cameras installed at different locations on a university campus. The cameras operate at different resolutions and at different distances from the monitored areas. Frames were captured at 30 fps and resized to 1,000 × 600 pixels before processing. We collected data over 5 recording sessions, 671 person tracks, and 24,721 person-crop images. All experiments were run on the free Google Colaboratory, which provides an NVIDIA Tesla T4 GPU (16 GB GDDR6), a 2-vCPU Intel Xeon processor (2.20 GHz), and approximately 12.7 GB of system RAM.

Module-level protocol. Each module was evaluated on a fixed test set, and all candidate models for that module were run on the same set with the same settings.

Detection and tracking: four detector–tracker configurations (YOLOv5s and YOLOv8s, each combined with DeepSort and ByteTrack) were run on the same video segment containing 165 annotated ground-truth person instances; the number of recovered instances and the total processing time were recorded.

Face detection: Haar Cascade, SSD, MTCNN, and RetinaFace were run on three test cases built from DTM output folders: a folder of 125 face images (Test Case 1), a mixed folder of 632 images of the same person containing 256 face and 376 non-face images (Test Case 2), and a folder of 36 non-face images (Test Case 3). We recorded detected faces, false positives, false negatives, and average time per image.

Face verification. We combined nine models with three distance metrics: Euclidean L2, Euclidean, and cosine. Each one is tested on a balanced dataset of 200 pairs, including 100 same-identity pairs and 100 different-identity pairs. The evaluation is based on

verification accuracy and the average decision time for each pair.

Person re-identification. Eleven configurations were evaluated with the standard query-gallery ReID protocol on data from the campus camera network, which yields 1,695 queries.

Evaluation of metrics. Detection coverage is the number of ground-truth person instances recovered out of 165.

For face detection. A false positive is an incorrectly classified face region, and a false negative is a face that isn't detected. Verification accuracy measures the percentage of pairs correctly identified as same or different. For ReID, Rank-k accuracy is the percentage of queries for which a correct match appears among the top k gallery results and mean Average Precision (mAP) is the mean, overall queries, of the average precision of the ranked gallery list. Processing time is reported as total run time for the DTM, and as mean time per image or per pair for the face modules.

Model selection criteria. For the DTM stage, we assess detection coverage across speed. Any person missing at this stage can't be recovered later in the pipeline. For face detection, we picked the detector that struck the best balance between false positives, false negatives, and processing time, since it acts as a routing gate (Section 3.2). For verification and ReID, we went with whichever configuration was most accurate (and for ReID, had the highest mAP), using processing time only to break ties between close contenders.

Integrated-system protocol. After model selection, the complete pipeline—ByteTrack-YOLOv5, MTCNN, SFace with Euclidean distance, and OSNet with triplet loss—was tested end-to-end on the campus recordings. Overall identification accuracy was calculated as the percentage of queries assigned to the correct identity, whether or not a face was visible. Of the 1,695 queries, 1,408 had no usable face and were processed using ReID, while 287 had usable facial information and were processed using face verification.

4. RESULTS AND DISCUSSION

The DTM evaluation was carried out using four detector-tracker combinations and a test set of 165 ground-truth person instances. The results showed noticeable differences between the four configurations, both in the number of people detected and in the consistency of the resulting tracking records.

The first configuration evaluated was YOLOv5 with DeepSort. It successfully recovered 104 of the 165

person instances and required 6 minutes and 24 seconds to complete the test, making it the slowest configuration. More importantly, the tracking results showed a repeated identity-mixing problem. In several cases, images belonging to different people were assigned to the same tracker ID, which meant that a single output folder could contain images of multiple individuals. Since the later stages of the system use these folders as individual identity records, this type of tracking error can affect the reliability of the subsequent identification process.

The YOLOv8 and DeepSort finished in 4 minutes 39 seconds, faster than YOLOv5 and DeepSort. It detected only 82 of 165 people, the lowest coverage. The tracking also had the same identity-mixing issue, where different individuals could be assigned to the same tracker ID, reducing reliability and suitability for later system stages.

The two ByteTrack configurations produced considerably better results. The association method used by ByteTrack was more effective in handling uncertain and low-confidence detections, which resulted in fewer tracking errors and higher detection coverage. YOLOv8 with ByteTrack detected 152 instances and completed the test in only 1 minute and 5 seconds, giving it the shortest processing time. YOLOv5 with ByteTrack recovered 160 of the 165 instances and took 2 minutes and 52.7 seconds, as shown in Figure 3. This was the highest detection coverage obtained in the evaluation.

The difference between the two ByteTrack configurations was therefore mainly a trade-off between processing time and detection coverage. YOLOv8 with ByteTrack was substantially faster, but it missed 13 instances that were detected by YOLOv5 with ByteTrack. In this system, a missed detection cannot be recovered by the later stages of the pipeline because the person is never passed to those stages in the first place. The additional processing time required by YOLOv5 with ByteTrack was about 108 seconds, while the alternative resulted in 13 additional missed instances. For this reason, YOLOv5 with ByteTrack was selected as the detection and tracking backbone for PAF-Net. It provided the highest detection coverage in the test while maintaining a reasonable processing time and produced a more suitable input for the identification modules that follow. The complete results are reported in Table 2, with Figures 4 and 5 providing additional visual comparisons.



Figure 3: Output Of Bytrack With YOLOv5

Table 2: The Result Of DTM

Detector	Tracker	165 Boxes	Time
YOLOv5	DeepSort	104	6m 24.4 s
YOLOv5	Bytrack	160	2m 52.7s
YOLOv8	DeepSort	82	4m 39.7 s
YOLOv8	Bytrack	152	1m 5.2 s

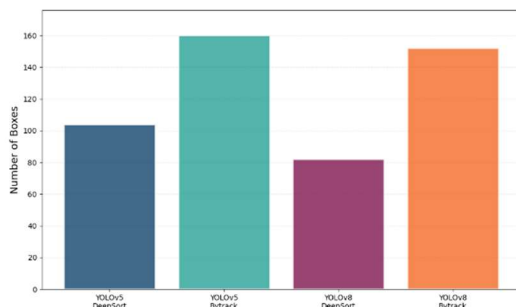


Figure 4: Detector-Tracking vs Number Of Boxes

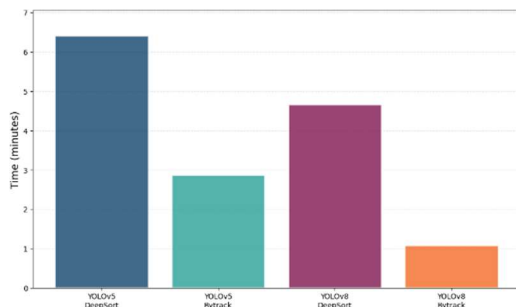


Figure 5: Detector-Tracking vs Processing Time

4.2 Face Detection and Recognition/Verification Models

4.2.1 Face detection model

Under the system, the model detects faces and indicates whether any are present in the captured image. Specifically, it classifies images with faces and passes them to the person re-identification phase. Each camera has a different resolution, so the model should work in both high and low resolutions. In an initial screening, all candidate detectors (Haar Cascade Classifier (OpenCV), SSD, MTCNN, Dlib, and RetinaFace) detected faces in high-resolution images. In contrast, only MTCNN, Haar Cascade

Classifier, and RetinaFace detected faces in low-resolution images, with varying degrees of success. Dlib was not carried forward, and the remaining four detectors were compared in the three test cases below. We tested the FD model on different case images from the DTM phase, as shown in the test cases below.

Test Case 1: Comparison of face detector models on a folder of 125 images, all containing faces, as shown in Table 3.

Test Case 2: Comparison of face detector models on a folder containing 632 mixed images, with 256 face images and 376 non-face images of the same person, as shown in Table 4.

Test Case 3: Comparison of face detector models on a folder containing 36 non-face images, as shown in Table 5.

In terms of accuracy and computation time, RetinaFace is the most accurate model, but it takes too long to detect a single face. It also predicts multiple boxes for the same face. In addition, the second-most accurate model, with reasonable runtime, is MTCNN. In contrast, the SSD and Haar Cascade models produce more false positives and false negatives than MTCNN. This makes MTCNN the best choice for the FD detector.

Table 3: Comparison Of Face Detector Models
Test Case 1

Model	Detecte d Faces	False Positiv e	False Negativ e	Averag e Time per Image
Haar_Casca de	52 faces	0	73	0.025
SSD	55 faces	5	70	0.050
MTCNN	75 faces	0	50	0.034
RetinaFace	128 faces	0	0	6.852

Table 4: Comparison Of Face Detector Models
Test Case 2

Model	Detecte d Faces	False Positiv e	False Negativ e	Averag e Time per Image
Haar_Cascad e	11 faces	1	245	0.013
SSD	182faces	53	203	0.066
MTCNN	1 faces	0	255	0.020
RetinaFace	265 faces	0	0	6.852

Table 5: Comparison Of Face Detector Models
Test Case 3

Model	Detect ed Faces	False Positiv e	False Negativ e	Average Time per Image
Haar Cascade	0 faces	0	4	0.011

SSD	37 faces	13	0	0.051
MTCNN	0 faces	0	0	0.007
RetinaFace	0 faces	0	0	7.558

Note on Table 5: Test Case 3 contains no faces, so every detection in this test case is a false detection. The Detected Faces column counts all bounding boxes returned by a detector, whereas the False Positive column counts the images in which at least one false face was reported; SSD's 37 boxes were therefore spread over 13 of the 36 images.

4.2.2 Face recognition/verification model

Face verification checks whether two facial images belong to the same person by comparing the feature representations extracted from each image in a common embedding space. Figure 6 shows this process using the VGG-Face model and cosine similarity. The two face images are first converted into feature vectors, and the similarity between these vectors is then calculated to determine whether the images likely belong to the same individual.

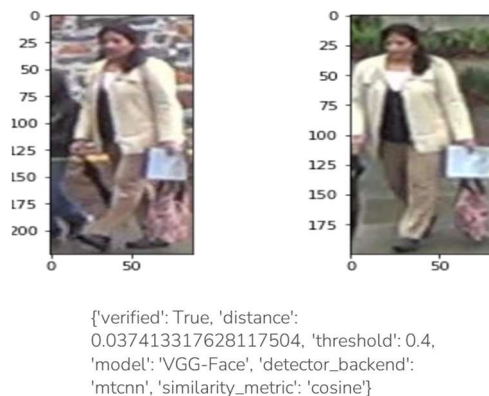


Figure 6: Example Of Face Verification Using The VGG-Face Model

To identify the most suitable model for the PAF-Net pipeline, we evaluated nine architectures: VGG-Face, FaceNet, FaceNet512, OpenFace, DeepFace, DeepID, ArcFace, SFace, and GhostFaceNet. Each model was tested against three distance metrics, Euclidean L2, Euclidean, and Cosine similarity, producing a comprehensive matrix of accuracy and speed measurements across twenty-seven distinct configurations. All models were assessed on a balanced test set of 200 sample pairs: 100 positive pairs drawn from images of the same individual, and 100 negative pairs drawn from images of different individuals. Verification accuracy and mean

decision time per pair were recorded for each configuration. Table 6 presents the full results, and Figures 7 and 8 visualize them.

Table 6: Comparison Of Face Recognition/Verification Models

Model	Euclidean_L2		Euclidean		Cosine	
	Success %	Avg Time	Success %	Avg Time	Success %	Avg Time
VGG-Face	90.36%	1.80s	90.36%	1.69s	89.16%	1.73s
Facenet	80.72%	1.27s	96.39%	1.27s	83.13%	1.27s
Facenet512	62.65%	1.28s	89.16%	1.25s	21.69%	1.31s
OpenFace	15.66%	0.69s	15.66%	0.67s	4.82%	0.65s
DeepFace	22.89%	5.50s	49.40%	5.73s	38.55%	5.56s
DeepID	2.41%	0.40s	3.61%	0.41s	2.41%	0.41s
ArcFace	95.18%	1.22s	62.65%	1.21s	95.18%	1.23s
SFace	6.02%	0.47s	99.80%	0.47s	10.84%	0.46s
GhostFaceNet	18.07%	1.11s	75.90%	1.09s	27.71%	1.11s

The majority of models performed poorly or inconsistently across distance metrics. OpenFace, DeepID, and SFace, when evaluated with Cosine similarity, all fell below 15% accuracy, making them unsuitable for any identity-sensitive application. DeepFace achieved a moderate recall of 49.40% under the Euclidean metric. Still, its average processing time of over 5.5 seconds per verification is prohibitive in a pipeline that may need to process hundreds of query images in a single session. GhostFaceNet and FaceNet512 showed mixed results depending on the distance metric used, with FaceNet512's Cosine accuracy dropping to just 21.69%, highlighting how sensitive some models are to the choice of similarity measure.

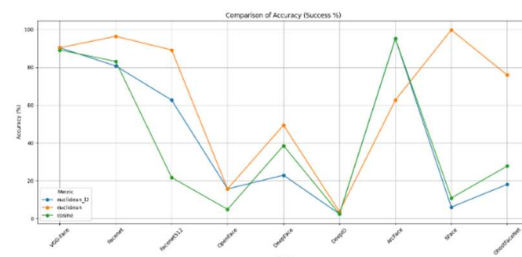


Figure 7: Comparison of Accuracy (Success %) vs Model

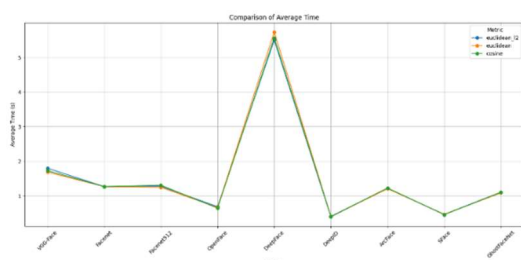


Figure 8: Average Time Comparison By Model

Three models showed noticeably stronger results than the others. VGG-Face gave similar results with both the Euclidean L2 and Euclidean metrics, reaching about 90% accuracy with an average verification time of approximately 1.75 seconds. The similar results across the two metrics indicate that its feature representations were relatively stable, although the accuracy remained limited for applications requiring highly reliable identification.

ArcFace achieved better results, reaching 95.18% accuracy with the Euclidean L2 metric while reducing the average verification time to 1.22 seconds. This represents an improvement in both accuracy and processing time compared with VGG-Face. The higher accuracy is consistent with the stronger separation between facial representations produced by ArcFace's angular-margin training approach.

SFace, however, produced the most compelling result. Using the Euclidean distance metric, it achieved a verification accuracy of 99.80% with an average decision time of just 0.47 seconds per pair, the highest accuracy and the fastest speed of any configuration tested, as shown in Figures 7 and 8. The combination of near-perfect accuracy and sub-second verification time makes SFace not only the strongest performer in this study, but a practically viable choice for real-time deployment. Accordingly, SFace with Euclidean distance was selected as the face verification model for the integrated PAF-Net system.

4.3 Person Re-Identification

Unlike the face-based modules, the Person Re-Identification (ReID) module relies on full-body appearance rather than facial information. It uses features such as clothing, body shape, posture, and gait to determine whether images captured by different cameras belong to the same person. Within the PAF-Net pipeline, the module is mainly used in two situations.

The first is to correct tracking errors. The folders generated by the DTM can sometimes

contain incorrect identity assignments. For example, images of two different people may be placed under the same tracker ID, or images of the same person may be divided between two different folders. The ReID module is used to identify these inconsistencies and correct them before the data is passed to the identification stage.

The second use is image-based retrieval. When a query image does not provide a usable face, the ReID module becomes the main identification method. It compares the person's full-body appearance with the records stored in the database and searches for matching appearances across the available camera views.

To determine which model was most suitable for these tasks, eleven ReID architectures were evaluated on the campus dataset. The tested models were ResNet50, DenseNet161, ResNet50-IBN-B, PCB-P6, SqueezeNet1_1, ShuffleNet, MobileNetV2_x1.4, OSNet_x1.0, OSNet-IBN_x1.0, OSNet-AIN_x1.0, and OSNet_x1.0 trained using triplet loss. The models were compared using Rank-1, Rank-5, Rank-10, Rank-20, and mean Average Precision (mAP). The complete results are given in Table 7.

There was a clear difference between the lighter models and the OSNet-based models. ResNet50 achieved a Rank-1 accuracy of 59.00% and an mAP of 33.10%, while SqueezeNet1_1 achieved a Rank-1 accuracy of 64.80%. These results were considerably lower than those obtained by the other models. ShuffleNet and DenseNet161 performed better, with Rank-1 accuracies of 91.70% and 92.10%, respectively. However, their mAP values were lower, at 79.20% and 88.50%, indicating that the correct identity was not consistently ranked near the top of the retrieved results.

The OSNet models produced the highest results overall. Standard OSNet_x1.0 achieved 99.97% Rank-1 accuracy and 99.90% mAP, while OSNet-AIN_x1.0 reached 99.80% for both measures. OSNet_x1.0 trained with triplet loss achieved a Rank-1 accuracy of 99.90% and the highest mAP, at 99.92%, among the tested models. Triplet loss helps organize the feature space by bringing images of the same person closer together while pushing apart images of different people. This behavior is particularly useful for the retrieval and

tracking-error correction tasks required by the ReID module.

Based on the results in Table 7, OSNet_x1.0 with triplet loss was used as the ReID backbone in the final PAF-Net system.

Table 7: Test Different P-ReID Models

Model	Testing: Rank-1	Testing: Rank-5	Testing: Rank-10	Testing: Rank-20	Testing: mAP
resnet50	59.00%	71.60%	75.80%	82.60%	33.10%
densenet161	92.10%	96.70%	99.60%	99.80%	88.50%
resnet50_ibn_b	99.80%	99.88%	99.96%	99.87%	98.70%
pcb_p6	97.90%	99.40%	99.79%	99.93%	98.10%
squeezenet1_1	64.80%	76.60%	82.60%	90.50%	49.10%
shufflenet	91.70%	96.10%	96.70%	97.90%	79.20%
mobilenetv2_x1_4	96.50%	99.40%	99.80%	99.80%	96.10%
osnet_x1_0	99.97%	99.96%	99.93%	99.98%	99.90%
osnet_ibn_x1_0	99.40%	99.48%	99.23%	99.96%	98.50%
osnet_ain_x1_0	99.80%	99.78%	99.97%	99.86%	99.80%
osnet_x1_0 with triplet loss	99.90%	99.97%	99.83%	99.88%	99.92%

4.4 PAF-Net System Integration

With the best-performing model identified at each stage of the pipeline, the final step was to integrate all modules into a single, functioning end-to-end system and verify that they operate cohesively under real-world conditions.

The integrated pipeline begins at the point of data ingestion. Video streams from the building's cameras are continuously fed into the detection and tracking module, which runs ByteTrack-YOLOv5 on each incoming frame. For each processed frame, the module identifies all persons present in the scene, crops their images, and stores them in individual tracking folders, one folder per tracker ID. Once enough data has accumulated over a given observation period, the cleaning process is applied to all folders, removing images that fall below the aspect ratio threshold or contain more than one person, as described in the methodology. What remains after cleaning is a structured collection of single-person image sets, each tied to a specific tracker ID, timestamp, and camera location.

At this stage, the face detection module checks each tracking folder to determine whether it contains any usable face images. Folders with at least one detected face are sent to the face verification module, where SFace, using Euclidean distance, compares the detected face with the identities stored in the

database. If no face is detected in a folder, the face verification stage is skipped, and the folder is sent directly to the OSNet-based ReID module. This module compares the person's full-body appearance with records from all available camera views to find a possible match. In either case, the resulting identity, whether confirmed or treated as a tentative match, is linked to the corresponding tracking folder. The database is then updated with the person's associated locations and timestamps.

Verification is required for both identification paths, and the ReID module is therefore part of the overall identification process rather than a separate component. In real surveillance footage, the same person may appear in face images in one camera and only full-body images in another. The integration layer handles these cases by comparing the resulting records and merging folders when they are identified as belonging to the same individual.

This process produces a single person record without duplicate entries. The consolidated table contains the available identity, location, and time information and is then used by the remaining parts of the system. Queries such as movement trajectories, time spent at different locations, and identity searches can therefore be generated from the same record. The complete architecture of the integrated PAF-Net system is shown in Figure 2.

4.5 Integrated System Performance

Table 8 summarizes the model selected for each module together with its module-level result. When these modules were run together on the campus recordings, PAF-Net achieved an overall identification accuracy of 98.7% within the integrated system for all five recording sessions. This system-level figure is not directly comparable with the module-level SFace result of 99.80%, which was obtained on isolated image pairs: in the integrated system, the verifier receives faces cropped automatically from real tracking folders, including small, blurred, or partially visible faces, and its outcome also depends on upstream detection and routing decisions, whereas the pair benchmark isolates the verifier from these effects.

Table 8: Summary Of Selected Models And Results

Module	Selected model	Module-level result
Detection and tracking	YOLOv5 + ByteTrack	160 of 165 instances; 2 m 52.7 s
Face detection	MTCNN	No false positives in all three test cases; 0.007–0.034 s per image
Face verification	SFace (Euclidean)	99.80% accuracy; 0.47 s per pair
Person ReID	OSNet_x1.0 + triplet loss	Rank-1 99.90%; mAP 99.92%
Integrated system	PAF-Net	98.7% overall identification

4.6 Discussion

Why ByteTrack outperforms DeepSort: Both ByteTrack configurations recovered more person instances than either DeepSort configuration (152–160 versus 82–104) and ran faster (Table 2). DeepSort associates only detections above a confidence threshold and computes a deep appearance descriptor for each of them. People who are partially occluded, far from the camera, or blurred therefore tend to be dropped or linked to the wrong track, which is consistent with both its lower coverage and the multiple-identities-per-tracker error observed in its output folders; the per-detection descriptor also adds processing time. ByteTrack additionally associates low-confidence boxes with existing tracks, which recovers these difficult cases. The comparison between detectors under the same tracker (160 versus 152 with ByteTrack, and 104 versus 82 with DeepSort) also shows that the newer detector is not automatically better on this data: YOLOv8 was considerably faster but produced fewer usable person boxes in this deployment. Because both detectors were used in their nano variants (YOLOv5s and YOLOv8s), this difference reflects the detector generation rather than model size.

Face detection behavior across test cases: The three test cases expose different failure modes (Tables 3–5). RetinaFace detected every face in Test Cases 1 and 2 and produced no detections in the non-face folder, but it reported 128 detections for 125 faces in Test Case 1 and 265 for 256 faces in Test Case 2; these extra detections are duplicate boxes on the same face rather than new faces. Its average time per image (6.85–7.56 s across the three test cases) is two to three orders of magnitude higher than that of the other detectors, which rules it out for a routing step that runs on every tracking folder. SSD reported false faces in the non-face folder of Test Case 3 and produced 53 false positives in Test Case 2, thereby sending face-absent folders into the verification

pathway. MTCNN was the only fast detector with no false positives in any test case. Its weakness is recall in Test Case 2, it detected only 1 of 256 faces, indicating that many faces in this folder fall below the quality threshold at which MTCNN operates reliably.

Within PAF-Net, this weakness is recoverable. A folder in which MTCNN finds no face is not discarded; instead, it is routed to the ReID module, which achieved Rank-1 accuracy above 99% on the same campus data, so the person can still be identified. A false positive, by contrast, directs a face-absent folder to the verifier, where no reliable match can be made. This asymmetry in error costs is why MTCNN was preferred over detectors with higher recall but more false positives, and it illustrates why module selection must consider the module's role in the pipeline rather than its stand-alone accuracy.

Face verification sensitivity to the distance metric: The most unexpected result in Table 6 is how strongly several models depend on the distance metric. SFace reached 99.80% with Euclidean distance but only 6.02% with Euclidean L2 and 10.84% with cosine; FaceNet512 dropped from 89.16% (Euclidean) to 21.69% (cosine); and ArcFace showed the reverse pattern, with 95.18% under Euclidean L2 and cosine but 62.65% under plain Euclidean distance. Accuracy far below 50% on a balanced pair set is worse than random guessing, which indicates that these failures originate in the decision threshold rather than in the embeddings themselves: each model's default threshold is calibrated for a particular metric and embedding scale, and when the metric changes, the threshold no longer separates same-identity from different-identity pairs. The practical implication is that the model, metric, and threshold must be selected together. Among the configurations with high accuracy, SFace with Euclidean distance combined the highest accuracy with the lowest time per pair (0.47 s, compared with 1.22 s for ArcFace and 1.80 s for VGG-Face).

Why OSNet performs best for re-identification: The OSNet family and ResNet50-IBN-B form a clear top group, with Rank-1 accuracy of at least 99.4% and mAP of at least 98.5%, while ResNet50 and SqueezeNet1_1 fall far behind (mAP of 33.10% and 49.10%) (Table 7). The large gap between Rank-1 and mAP for DenseNet161 and ShuffleNet shows that these models often rank one correct image first but place the remaining correct images lower in the list. This matters for PAF-Net

because the ReID module must retrieve all folders belonging to a person to merge them, not only the top match. OSNet's omni-scale design captures both fine details, such as clothing texture, and whole-body shape, which suits campus footage in which the same person appears at very different sizes across cameras. The strong result of ResNet50-IBN-B (99.80% Rank-1) compared with the plain ResNet50 (59.00%) is consistent with the benefits of instance normalization in mitigating color and illumination differences across cameras. Training OSNet_x1.0 with triplet loss gave the highest mAP (99.92%); because mAP reflects the quality of the whole ranked list, it was given priority over the small Rank-1 difference with the standard OSNet_x1.0 (99.90% versus 99.97%).

5. CONCLUSION, LIMITATIONS, AND FUTURE WORK

This paper presented the People Analytics Framework (PAF-Net), an end-to-end surveillance pipeline designed to identify and track individuals across multiple cameras in closed, private-network environments. The system integrates four core modules — person detection and tracking, face detection, face verification, and person re-identification — into a unified pipeline capable of answering the fundamental surveillance questions of who, where, and when for every individual who appears within the monitored area. By combining face-based and full-body appearance-based identification pathways, PAF-Net addresses a limitation that most existing systems leave unresolved: the inability to identify a person when the camera does not capture their face.

The system was deployed and tested using four surveillance cameras located on a university campus. It achieved an overall identification accuracy of 98.7%. The system was also able to identify individuals when their faces were only partly visible or not visible at all. In these cases, the ReID module provided an alternative identification path based on full-body appearance. The final configuration used YOLOv5 with ByteTrack for detection and tracking, MTCNN for face detection, SFace for face verification, and OSNet with triplet loss for person re-identification. Each component performed well in its individual evaluation and worked together as part of the complete system.

Key findings. Four main findings were obtained from the experiments. First, the choice of tracker had a greater effect on detection and tracking results than the detector version did. ByteTrack provided better

coverage and faster processing than DeepSort, with YOLOv5 and ByteTrack recovering 160 out of the 165 person instances in the test set. Second, when the face detector is used primarily to determine which identification path to follow, false positives can be particularly problematic. Missed faces can still be handled through the ReID path, whereas an incorrect face detection can send a tracking folder to the wrong verification stage. Among the tested fast detectors, MTCNN produced no false positives in the evaluated cases. Third, face verification performance varied depending on the model, distance metric, and threshold used in combination. SFace with Euclidean distance achieved 99.80% accuracy with an average decision time of 0.47 seconds per pair. Fourth, OSNet trained with triplet loss produced the highest mAP among the tested ReID configurations, reaching 99.92%. This result is particularly relevant to the folder-merging and image-retrieval tasks, where the position of the correct identity within the ranked results is important.

Implications: For practitioners, the results show that a closed-area surveillance system can identify people who never face a camera by adding an appearance-based pathway, without sending video outside the organization's network. They also show that model choices should be made on data from the target site and in light of the role each module plays in the pipeline, rather than on public benchmark rankings alone. For researchers, the study shows the value of evaluating recognition components as parts of a routed system, in which the cost of an error depends on what the next module can recover.

Two main challenges came up during the development and evaluation of PAF-Net. The first was the data collection process. Since the cameras were placed at different distances and operated at different resolutions, the quality of the captured images varied across cameras. This variation affected the consistency of detection and identification throughout the system.

The second challenge was data annotation. Matching and labeling the same individuals across multiple camera views required significant manual effort. This was particularly challenging for the ReID module because the training data needed to maintain the same identity labels across different cameras. Even small errors in matching an individual between camera views could affect the accuracy of the ReID results.

Limitations: The study has several limitations that bound the generality of its results. Data were

collected at a single university campus using only four cameras. While model selection was based on actual testing, if the data changes, all candidate models would need to be retested and re-ranked, since accuracy, image count, and lighting directly affect performance. Test set size. Some module-level test sets are relatively small. The DTM test set has 165 person instances, and the face verification test set has 200 pairs. This limits the statistical precision when comparing models with close performance. Confidence intervals were not reported, and the tests were not repeated—ReID accuracy saturation. ReID accuracy is close to saturation for several models. This suggests that the campus test data is less challenging than public benchmarks. Long-term scenarios, such as clothing changes or re-appearance on different days, were not evaluated. Performance under continuous operation. The integrated pipeline processes accumulated tracking folders in batches, which we didn't evaluate. End-to-end latency and resource usage under continuous live streaming on edge hardware still need to be measured.

Future work will focus on addressing these challenges in two main areas. The first is reducing the amount of manually labeled data required by the system. Unsupervised domain adaptation and semi-supervised learning will be investigated to improve PAF-Net's ability to generalize to new environments, particularly when cameras, lighting conditions, and scene layouts differ from those used during development.

The second area concerns privacy and deployment. Privacy-preserving methods and edge-based implementations will be considered to make the system more suitable for practical use and to reduce some of the privacy concerns associated with automated surveillance. These improvements could help make PAF-Net more adaptable to different environments and easier to deploy in real-world settings.

Several additional experiments are also needed to evaluate the limitations of the current system. These include testing PAF-Net at different locations and with different camera arrangements, increasing the size of the evaluation sets, and reporting confidence intervals for the measured results. The ReID component should also be tested on public benchmark datasets and under more challenging conditions, such as long-term monitoring, where a person's clothing may change. Another area for further study is the calibration of verification

thresholds for the different distance metrics used in the face verification stage.

End-to-end processing time should also be measured on edge devices using live video streams rather than only offline recordings. Finally, the current rule-based decision between the face and ReID pathways could be replaced with a learned fusion approach. Instead of selecting one pathway based on whether a face is detected, the system could combine the face-verification and ReID scores. This would allow information from both sources to contribute to the final identity decision, which may be particularly useful when only part of a person's face is visible.

6. DECLARATIONS

Ethical considerations: The video data used in this study were recorded on the authors' university campus within a closed, private network and were used only for research purposes.

Conflict of interest: The authors declare no conflicts of interest.

Funding: This research was conducted as part of research at the Department of Systems and Computer Engineering, Al-Azhar University. The authors gratefully acknowledge the university's institutional support. No external funding was received for this study.

Declaration on the use of AI in writing: AI-assisted tools were used for language and grammar editing only. AI generated no content, results, or references.

REFERENCE:

- [1] C. B. Murthy et al., "Investigations of Object Detection in Images/Videos Using Various Deep Learning Techniques and Embedded Platforms: A Comprehensive Review", *Applied Sciences*, Vol. 10, No. 9, May 2020, doi: 10.3390/app10093280.
- [2] D. Fleet et al. (Eds.), *Computer Vision – ECCV 2014*, Vol. 8690, Springer International Publishing, 2014, doi: 10.1007/978-3-319-10605-2.
- [3] L. Jiao et al., "A Survey of Deep Learning-Based Object Detection", *IEEE Access*, 2019, doi: 10.1109/ACCESS.2019.2939201.
- [4] Z. Zou et al., "Object Detection in 20 Years: A Survey", *arXiv preprint arXiv:1905.05055*, May 2019.

- [5] S. S. Vasekar, S. Patil, and R. Ban, "Object Tracking in Video Surveillance", *International Journal of Research Publication and Reviews (IJRPR)*, 2023, www.ijrpr.com.
- [6] N. Wojke, A. Bewley, and D. Paulus, "Simple Online and Real-time Tracking with a Deep Association Metric", *arXiv preprint arXiv:1703.07402*, 2017.
- [7] M. Abouelyazid, "Comparative Evaluation of SORT, DeepSort, and ByteTrack for Multiple Object Tracking in Highway Videos", *International Journal of Sustainable Infrastructure for Cities and Societies*, Vol. 8, No. 11, Nov. 2023, pp. 42-52.
- [8] D. R. Patrikar and M. R. Parate, "Anomaly Detection Using Edge Computing in Video Surveillance Systems: A Review", *International Journal of Multimedia Information Retrieval*, Vol. 11, No. 2, June 2022, pp. 85-110, doi: 10.1007/s13735-022-00227-8.
- [9] X. Li, Z. Yang, and H. Wu, "Face Detection Based on Receptive Field Enhanced Multi-Task Cascaded Convolutional Neural Networks", *IEEE Access*, Vol. 8, 2020, pp. 174922-174930, doi: 10.1109/ACCESS.2020.3023782.
- [10] J. Deng et al., "ArcFace: Additive Angular Margin Loss for Deep Face Recognition", *arXiv preprint arXiv:1801.07698*, Jan. 2018, doi: 10.1109/TPAMI.2021.3087709.
- [11] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A Unified Embedding for Face Recognition and Clustering", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 815-823, doi: 10.1109/CVPR.2015.7298682.
- [12] X. Wang et al., "A Survey of Face Recognition", *arXiv preprint arXiv:2212.13038*, Dec. 2022.
- [13] R. J. Hassan and A. M. Abdulazeez, "Deep Learning Convolutional Neural Network for Face Recognition: A Review", 2021, doi: 10.5281/zenodo.4471013.
- [14] K. Zhou et al., "Omni-Scale Feature Learning for Person Re-Identification", *arXiv preprint arXiv:1905.00953*, May 2019.
- [15] V. D. Nguyen, S. Mirza, A. Zakeri, A. Gupta, K. Khaldi, R. Aloui, P. Mantini, S. K. Shah, and F. Merchant, "Tackling Domain Shifts in Person Re-Identification: A Survey and Analysis", *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2024, arXiv:2404.17568.
- [16] C. B. Nalty et al., "A Brief Survey on Person Recognition at a Distance", *arXiv preprint arXiv:2212.08969*, Dec. 2022.
- [17] B. R. Ardabili et al., "Understanding Ethics, Privacy, and Regulations in Smart Video Surveillance for Public Safety", *arXiv preprint arXiv:2212.12936*, Dec. 2022.
- [18] O. Elharrouss, N. Almaadeed, and S. Al-Maadeed, "A Review of Video Surveillance Systems", *Journal of Visual Communication and Image Representation*, May 2021, doi: 10.1016/j.jvcir.2021.103116.
- [19] M. Zahrawi and K. Shaalan, "Improving Video Surveillance Systems in Banks Using Deep Learning Techniques", *Scientific Reports*, Vol. 13, No. 1, Dec. 2023, doi: 10.1038/s41598-023-35190-9.
- [20] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN", *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2961-2969, doi: 10.1109/ICCV.2017.322.
- [21] S. Angadi, S. Nandyal, and P. Doddappa, "Review on Object Detection and Tracking in Video Surveillance", *International Journal of Advanced Research in Engineering and Technology*, Vol. 11, No. 9, 2020, pp. 1033-1042, doi: 10.34218/IJARET.11.9.2020.102.
- [22] A. Singh et al., "Automation of Surveillance Systems Using Deep Learning and Facial Recognition", *International Journal of System Assurance Engineering and Management*, Vol. 14, Mar. 2023, pp. 236-245, doi: 10.1007/s13198-022-01844-6.