Multi-Camera Person Re-Identification for Hospital Process Tracking

by

Kai Wang

Supervisor: Andrew Brown

April 7th, 2025

B.A.Sc. Thesis

Division of Engineering Science
UNIVERSITY OF TORONTO

Abstract

This thesis presents a hallway-based multi-camera person Re-Identification (Re-ID) system for estimating patient time spent at key stages in the CT imaging process, such as waiting, changing and scanning. The system uses YOLOv8 for detection, DeepSORT for per-camera tracking, and a global matcher to assign consistent identities across different camera views using visual embeddings from PCB, OSNet, and TransReID.

Real hospital hallway footage was used for evaluation. OSNet demonstrated the best balance of accuracy and speed, while PCB and TransReID faced challenges under occlusions or low-resolution views. A key finding is that continuous tracking may be unnecessary — Re-ID at entry and exit points alone could meet clinical needs with lower complexity and reduced risk of identity errors.

This work highlights the potential of lightweight, privacy-conscious Re-ID pipelines for real-time patient flow monitoring in hospital environments.

Acknowledgements

I would like to thank my thesis supervisor, Dr. Andrew Brown, and his team,
for their continuous support and guidance throughout this project.
I also wish to thank the Medical Imaging staff at St. Michael's Hospital, Toronto,
for their cooperation during data collection.

Table of Contents

List of Figures and Tables

1. Introduction

1.1 Motivation

Diagnostic medical imaging services are a cornerstone of modern healthcare, but they rely heavily on tightly coordinated resources — equipment, staff, space, and time. Delays or inefficiencies in this pipeline can reduce the overall quality of patient care. In Ontario, for example, only 58% of adults received timely CT scans as of June 2024, falling short of provincial targets [1]-[4]. To address this growing concern, there is a pressing need for real-time workflow analytics that can help administrators proactively detect and resolve bottlenecks in radiology operations.

1.2 Prior Art

Historically, several methods have been proposed to monitor imaging workflows. A cost-effective but error-prone approach involves having technologists manually log patient activity and room usage. However, due to human error and documentation fatigue, these logs are often incomplete or imprecise. A more reliable solution is to perform external observation, such as hiring someone to monitor and timestamp patient transitions. This has been done either in person or post hoc through surveillance video analysis [3]. While accurate, such methods are labor-intensive, costly, and unsustainable in the long term, especially in high-volume clinical environments.

More recently, researchers have turned to computer vision for passive and scalable workflow monitoring. At St. Michael's Hospital, Toronto, for instance, the YOLO-AR system [5] combined YOLOv8-based segmentation with action recognition to identify procedural stages during CT scans. Their solution produced detailed insights into the interactions between patients, staff, and equipment, all while ensuring privacy through non-facial vision models. While that system focused on in-room action analysis, it highlights the broader potential of vision-based tools to optimize radiology workflows.

1.3 Goal

This thesis addresses a related but distinct challenge: tracking patients across multiple hospital zones (e.g., waiting room, changing area, CT scan room) using ceiling-mounted hallway cameras. The system aims not to monitor detailed actions but to determine when and where patients enter and exit key locations, enabling the estimation of time spent at each step in the process. This timeline can help identify workflow inefficiencies such as prolonged wait times or bottlenecks in scan preparation. While facial recognition is technically permitted, it was intentionally avoided in this work to reduce privacy concerns and improve generalizability across settings. Instead, the system relies on non-facial visual cues like clothing or body shape for identification. Although a lightweight tracking module is currently integrated to maintain person continuity within each camera view, one key insight from development is that long-range tracking across time and space can introduce accumulating errors. Therefore, a core direction for future improvement is to remove continuous tracking altogether and focus on discrete Re-ID events at critical room boundaries, allowing for accurate timestamping and more robust, simpler system behavior.

2. Background

2.1 Person Re-Identification Fundamentals

Person Re-Identification (Re-ID) refers to assigning consistent identities to individuals captured across different frames or camera views, often without temporal or spatial continuity. Unlike object tracking, which relies on continuous motion estimation within a single camera stream, Re-ID systems are designed to recognize the same person reappearing in potentially non-overlapping views, possibly after a period of disappearance. The standard Re-ID pipeline involves three components: person detection, feature extraction using a deep neural network, and similarity-based matching using metrics like cosine distance or Euclidean distance between embeddings [6].

Re-ID is widely recognized as a challenging task due to significant intra-class variation and inter-class similarity. The same individual may look drastically different due to

- changes in viewpoint (e.g., front-facing vs back-facing),
- illumination (e.g., hallway shadows vs direct lighting),
- occlusion (e.g., partially blocked by objects or other people),
- and especially appearance changes, such as putting on hospital gowns.

These appearance-altering events can confuse even high-performing Re-ID models, which are typically trained on public datasets like Market-1501 or MSMT17 [7][8]. Moreover, the embeddings generated by Re-ID models are not always invariant to camera resolution or angle distortions, which can lead to identity mismatches.

Despite these challenges, Re-ID has seen growing adoption in multi-camera surveillance, where fixed spatial relationships between cameras may not exist. In such cases, Re-ID offers a way to reconnect individuals across locations purely through appearance-based features, without requiring trajectory continuity or camera calibration. In hospital environments, this makes Re-ID a promising candidate for privacy-aware patient tracking, especially when facial recognition is discouraged or infeasible.

2.2 Detection + Tracking vs Re-ID Approaches

Conventional person-tracking pipelines in computer vision typically combine a real-time object detector with a multi-object tracking (MOT) algorithm. The detector (e.g., YOLO [9]) identifies person bounding boxes in each frame, while the tracker (e.g., DeepSORT [10]) associates these detections over time using both motion and appearance cues. This works well in single-camera settings with continuous visibility, as the temporal information (e.g., frame-by-frame location prediction via a Kalman

filter) allows for relatively stable ID assignment, even during brief occlusions or overlaps.

However, such detection–tracking pipelines face substantial limitations in multi-camera scenarios, particularly when cameras have non-overlapping fields of view. Trackers rely heavily on temporal continuity and spatial proximity, both of which break down when a person disappears from one camera and later reappears in another. In such cases, identity must be reassigned not through motion extrapolation, but through appearance-based reasoning — a task that conventional trackers are not designed to handle robustly. Furthermore, long-duration tracking over dozens or hundreds of frames may lead to ID drift, where a person's assigned identity gradually deviates due to compounding matching errors, missed detections, or occlusion events.

To address these limitations, Person Re-ID has emerged as an alternative approach. Rather than relying on temporal linkage, Re-ID treats every detection independently and seeks to match them via learned visual embeddings. This makes Re-ID particularly suitable for settings where people reappear after a delay or change appearance, such as in hospital settings, where patients may temporarily leave the field of view or appear in different clothing (e.g., hospital gowns). In this project, Re-ID is used to assign consistent global IDs across multiple hallway cameras, overcoming the constraints of short-term tracker memory and enabling timeline estimation for key patient steps. While a lightweight tracker (DeepSORT) is still used to stabilize local ID assignment within each camera, Re-ID plays the central role in linking those local tracks into a unified patient identity across the entire hospital workflow.

2.3 Re-ID Models

To extract robust feature embeddings from detected person crops, this project evaluates three state-of-the-art Re-ID models: **PCB**, **OSNet**, and **TransReID**. Each model offers a different trade-off between performance, complexity, and deployability, making them suitable for experimentation in constrained hospital environments where compute, privacy, and reliability all play critical roles.

2.3.1 Part-based Convolutional Baseline

Part-based Convolutional Baseline(PCB), proposed by Sun et al. [11], is a person Re-ID architecture that improves discriminative performance by dividing the input person image into six horizontal stripes. Each stripe is processed independently to learn localized features, which are then concatenated to form the final embedding vector. This approach was motivated by the observation that different body parts carry complementary visual cues, and pooling them separately allows the model to better handle pose variation and partial occlusion.

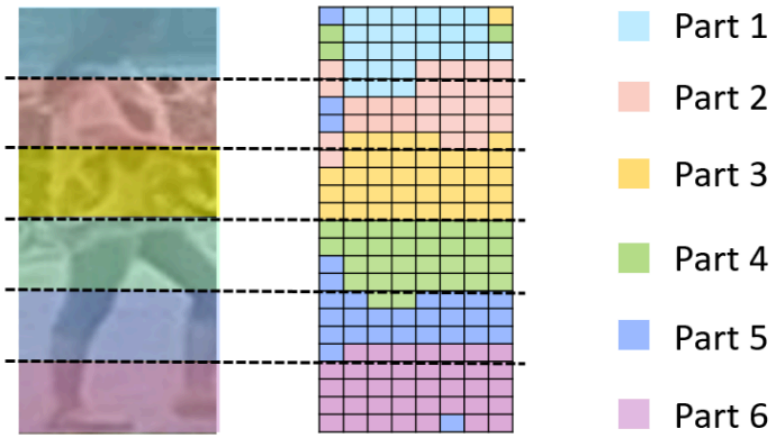


Figure 1. Visualization of striping methodology in PCB[11].

The image is split into six horizontal parts, each pooled separately for feature extraction.

The PCB design has been shown to outperform global feature extractors, especially in settings where part-level alignment is strong and occlusion is minimal. However, it

can be sensitive to misalignment and viewpoint variation, and relies on a fixed stripe partitioning scheme. In crowded or unconstrained scenes, this rigidity can introduce feature noise when one stripe contains pixels from multiple individuals or background clutter.

Despite its age, PCB remains a well-understood and interpretable baseline in the Re-ID literature, and continues to be used in controlled benchmarks and ablation studies.

2.3.2 Omni-Scale Network

Omni-Scale Network(OSNet), proposed by Zhou et al. [12], was designed specifically for person re-identification in real-world, unconstrained settings. The key innovation in OSNet is its ability to extract features at multiple perceptual scales(*Appendix A*) simultaneously — a capability the authors term "omni-scale feature learning." This is achieved using a specialized module called the OSBlock, which fuses outputs from multiple parallel convolutional paths of varying receptive field sizes (e.g., 1×1, 3×3, 5×5).

Whereas traditional convolutional networks extract features using fixed-scale filters, OSNet processes input through a hierarchy of scales within each residual block, enabling the network to capture fine-grained local details (e.g., textures, shoes) and global body structure (e.g., silhouette, clothing patterns) at the same time. These multi-scale features are then fused using a learned gate mechanism that dynamically weights the contribution of each scale, allowing the network to adapt to input variation.

This omni-scale design is particularly valuable for hospital hallway surveillance, where patients may appear at different distances from the camera, under varying lighting, and with differing visibility of body parts. For instance, fine-scale features may be lost when a patient appears far from the camera, but coarse-scale features —

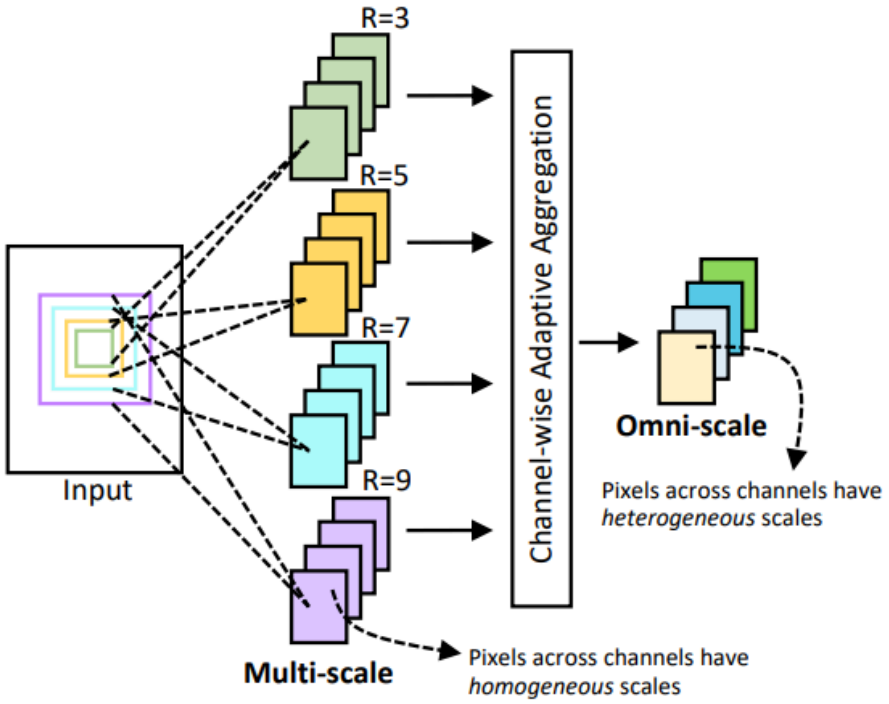


Figure 2. Illustration of OSNet's omni-scale feature learning architecture[12]

like general body shape or color distribution — may still be informative. Conversely, when patients are nearby, more detailed texture cues (e.g., footwear) become important. OSNet's ability to learn and adapt across these scales improves its robustness in such dynamic conditions.

In addition to its representational flexibility, OSNet is also highly efficient. The architecture was explicitly designed for lightweight inference, making it well-suited for deployment on edge devices like the NVIDIA Jetson used in this project. Despite its compact size, OSNet achieves competitive accuracy on standard Re-ID benchmarks like Market-1501 and MSMT17, outperforming many heavier models in both speed and accuracy trade-offs.

Due to its lightweight design and robustness across scale and appearance variation, OSNet has been widely adopted in edge-deployable Re-ID systems, especially in resource-constrained surveillance settings. Its ability to perform well under varying visibility and spatial resolution conditions makes it a strong candidate for scenarios such as hospitals, where subjects appear under different viewpoints and distances from ceiling-mounted cameras.

2.3.3 Transformer-Based Re-Identification

Transformer-Based Re-Identification(TransReID), introduced by He et al. [13], represents a new wave of transformer-based architectures adapted for person re-identification. Building on the success of Vision Transformers (ViTs) in image recognition, TransReID adapts the ViT backbone to better handle the challenges unique to Re-ID, including viewpoint variation, occlusion, and part misalignment.

At its core, TransReID replaces traditional CNN backbones with a ViT structure that divides the input image into a grid of patches and models long-range dependencies using self-attention. This global receptive field allows the model to reason over complex spatial relationships that CNNs might miss. However, vanilla ViTs struggle with Re-ID due to a lack of spatial bias and difficulty in handling small-scale variations. To address this, TransReID introduces several Re-ID-specific enhancements, including:

- Jigsaw Patch Module (JPM): Adds local structure awareness by rearranging patch positions and forcing the model to learn spatial relationships.

- Camera-aware Position Embedding (CAPE): Introduces positional biases that account for different camera viewpoints, improving cross-camera matching.

- Part Attention Module: Encourages the model to focus on distinct body regions, helping recover from occlusions or appearance inconsistencies.

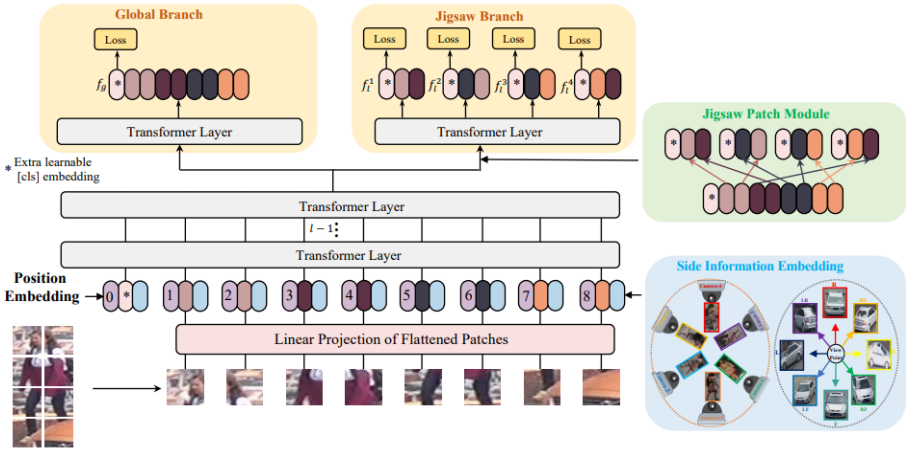


Figure 3. Architecture of TransReID and its specialized modules.[13]

These components make TransReID more robust to cross-view appearance change, a key challenge in person Re-ID. Empirically, TransReID has achieved state-of-the-art performance on large-scale Re-ID datasets like Market-1501, DukeMTMC, and MSMT17, outperforming both CNN-based models and earlier ViT-based baselines.

While TransReID offers strong accuracy, it is also computationally expensive compared to models like OSNet or PCB. The transformer backbone requires more memory and compute, making it less ideal for real-time inference on edge devices without GPU acceleration. However, it serves as a powerful benchmark to assess the upper bound of Re-ID performance, especially in controlled evaluation settings where compute constraints are relaxed.

2.3.4 Model Comparison

The three Re-ID models discussed — PCB, OSNet, and TransReID — represent different philosophies in person re-identification design, each with trade-offs in performance, complexity, and deployment feasibility.

PCB offers a structured and interpretable part-based design but lacks flexibility in real-world environments where pose variation and occlusion are common. OSNet strikes a balance between robustness and efficiency by capturing omni-scale features in a lightweight architecture, making it highly suitable for resource-constrained edge devices. TransReID, on the other hand, leverages transformer-based global attention mechanisms and achieves top-tier accuracy, but its higher computational cost can pose challenges for real-time applications, particularly on embedded hardware like the NVIDIA Jetson.

The following table summarizes the key characteristics of each model:

Model	Architecture Type	Key Idea	Strengths	Limitations	Edge Suitability
PCB	CNN, part-based	Horizontal Stripe Pooling	Interpretable, simple, can avoids facial focus	Rigid part division, sensitive to misalignment and occlusion	Moderate
OSNet	CNN, omni-scale (all scales)	Multi-scale conv + fusion	Compact, fast, good performance, suitable for real-time use	Slightly lower peak accuracy than larger models	High
Trans ReID	Transformer-based	Global self-attention + part-aware	High accuracy, robust to occlusion and viewpoint shift	High inference cost, memory-heavy	Low

Table 1. Comparison of three Re-ID models evaluated in this project.

2.4 Related Work in Hospital Settings

Computer vision has increasingly been explored as a tool to improve operational efficiency and workflow monitoring in clinical environments. In particular, radiology and diagnostic imaging departments are resource-intensive areas where workflow optimization can have a substantial impact on cost, throughput, and patient experience. Several prior studies have investigated how video-based systems can aid in measuring process durations, identifying bottlenecks, and improving resource allocation.

A common approach to workflow analysis in hospitals relies on manual logging by technologists or staff, where key steps such as patient arrival, scan start, and exam completion are time-stamped. While cost-effective, this approach is vulnerable to human error, inconsistent reporting, and incomplete records. A more reliable alternative involves external post hoc video analysis, as seen in prior work such as [3], where observers manually reviewed surveillance recordings to extract workflow timings. However, this method is time-consuming, labor-intensive, and not scalable for real-time feedback or large patient volumes.

Some hospital systems have experimented with automated rule-based tracking using ceiling-mounted cameras and simple motion detection algorithms. For example, motion thresholds or room entry events are used to infer patient presence. While useful for coarse occupancy sensing, these systems generally lack the ability to maintain identity over time, especially when patients appear in multiple disjoint camera zones or undergo appearance changes (e.g., by changing into hospital gowns).

Facial recognition could, in theory, offer a more precise identity signal, but its use in healthcare settings raises significant privacy and ethical concerns, particularly under regulatory frameworks such as HIPAA or PHIPA(privacy laws). Consequently, there has been growing interest in appearance-based person Re-ID methods that rely on body-level features like clothing, gait, or footwear to match individuals across views, though such systems have seen limited deployment in real-world clinical environments.

These limitations highlight a gap in existing hospital-focused vision systems: the lack of identity-preserving, privacy-conscious tracking that can operate across multiple, non-overlapping hallway cameras. While appearance-based Re-ID offers a promising solution, its application to hallway footage, where patients may change clothing, walk with others, or appear far from the camera, remains underexplored in the literature. This thesis responds to that gap by examining whether state-of-the-art Re-ID models can support such systems in a way that is both scalable and privacy-aligned.

3. Methods

3.0 Preprocessing

Before detection and tracking can occur, each video stream undergoes a set of preprocessing steps to ensure synchronization, spatial focus, and model compatibility. These steps are essential for creating a clean input pipeline for the downstream detection and Re-ID models.

3.0.1 Camera Setup and Synchronization

Data was collected using three ceiling-mounted hallway cameras at St. Michael's Hospital, Toronto, covering critical zones around the CT scan room and changing room. Although all cameras were launched with the same codebase, slight timing offsets were observed between streams. To correct for this, the initial few frames were manually trimmed so that all videos start at the same patient movement moment. This synchronization step is crucial for downstream global ID matching and time-based timeline reconstruction.

All videos were recorded at 30 FPS in H.264 format, providing sufficient temporal granularity for real-time Re-ID. Data was processed locally on an NVIDIA Jetson device, reflecting the project's aim to run entirely on edge hardware in a privacy-preserving manner.

3.0.2 Region of Interest (ROI) Design

To reduce false detections and focus on meaningful patient movement, the ROI was addressed at the hardware level by physically placing and orienting each camera to cover only the zones of interest. (*Appendix B*)

- Camera 1 provided a view of a larger transitional area, covering the doorway of CT scan room and the changing room.
- Camera 2 was positioned to capture the hallway leading directly to the changing room.
- Camera 3 covered the entrance to the CT scan room.

This physical ROI control helped limit visual clutter and reduce noise from irrelevant activity, such as staff movement or side room interactions, ultimately making the detection and Re-ID stages more focused and efficient.

3.0.3 Frame Resize

All frames are resized to match the input requirements of the chosen Re-ID model. For example, PCB expects person crops of size 128×384, while OSNet and TransReID accept different dimensions based on their pretrained configuration.

3.1 System Overview

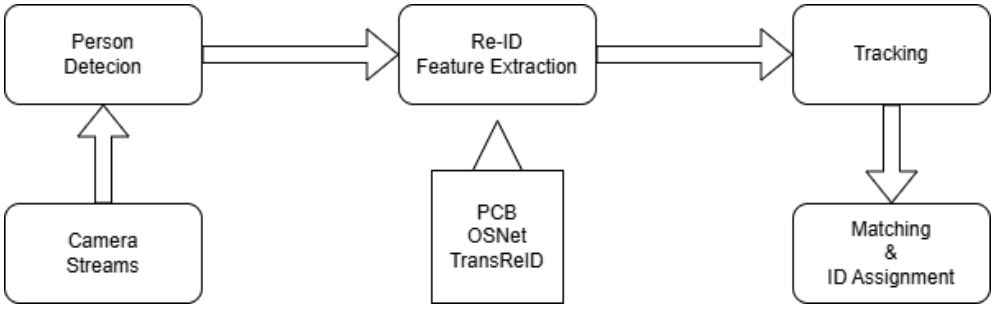
The system is designed to estimate patient timelines in CT scan workflows by determining when a patient enters and exits key hospital zones, such as the changing room and CT scan room, using hallway surveillance footage. The current implementation attempts to maintain continuous tracking of individuals throughout their journey by combining short-term tracking within each camera with appearance-based Re-ID across cameras. This enables both intra-camera identity assignment and cross-camera identity consistency, which are necessary for timeline estimation.

The architecture is modular, consisting of five main components:

1. **Detection:** Each frame is processed by a YOLOv8 detector to identify bounding boxes around people.

2. **Re-ID Feature Extraction:** For every detected person, a visual embedding is extracted using a state-of-the-art Re-ID model (PCB, OSNet, or TransReID).

3. **Local Tracking:** Within each camera stream, DeepSORT uses motion and appearance cues to assign short-term tracking IDs to individuals.



4. **Global Matching:** A feature-based matcher compares appearance embeddings across cameras and over time to assign a unified global ID.

5. **Timeline Estimation (future work):** Entry/exit events tied to each global ID will be used to estimate time spent in each room, helping identify bottlenecks in clinical workflows.

Each camera operates independently for detection and local tracking. Only lightweight feature vectors are passed to the global matcher, enabling edge-friendly deployment on devices like NVIDIA Jetson. The system is capable of handling reappearances after clothing changes, partial occlusion, or movement between disjoint camera views.

3.2 Detector with YOLO

The first stage of the pipeline involves detecting people in each video frame using the YOLOv8 (You Only Look Once, version 8) object detection model. YOLOv8 is a state-of-the-art real-time object detector developed by Ultralytics, known for its balance between accuracy and speed, making it suitable for edge device deployment.

In this project, YOLOv8 is used to generate person bounding boxes in each frame across all three camera streams. Only detections corresponding to the “Person” class (class index 0 in the COCO dataset) are retained. This filtering is critical to minimize false positives and reduce the number of unnecessary feature extractions downstream.

3.2.1 Model Details

- **Model Version:** yolov8n.pt (nano version) or yolov8x.pt (extra-large version)
- **Framework:** Ultralytics YOLOv8
- **Input Format:** Raw RGB video frames
- **Inference Output:** Bounding box coordinates (x1, y1, x2, y2) and confidence scores

Multiple YOLOv8 variants were tested during development. While the yolov8x model offers higher detection accuracy due to its larger capacity, it is computationally

expensive and not ideal for real-time performance on devices like the NVIDIA Jetson. The yolov8n (nano) version was ultimately selected for its efficiency and faster inference speed during deployment testing.

3.2.2 Detection Frequency and Frame Skipping

To balance computational cost with temporal coverage, detections are not run on every frame. Instead, a frame-skipping strategy is applied (e.g., running detection every 5 frames). This significantly reduces the load on the Jetson while still providing enough temporal granularity for downstream tracking and Re-ID tasks.

3.2.3 Bounding Box Refinement and Filtering

To further improve the quality of features extracted from each detection, the bounding boxes returned by YOLOv8 are refined in two key ways: Padding Shrink & Non-Maximum Suppression(NMS).

A small amount of padding is removed from each detection to eliminate irrelevant background. This helps focus the cropped image on the person's body, reducing noise from walls, equipment, or shadows that might otherwise interfere with embedding quality.

pad = 0.05 # 5% padding removed from all sides

This 5% inward adjustment tightens the bounding box on all sides, which is especially beneficial in confined hospital spaces where the background is often cluttered and similar across frames.

YOLOv8 applies NMS internally to remove overlapping bounding boxes that likely correspond to the same person. NMS retains only the most confident detection when multiple boxes overlap beyond a threshold, reducing redundancy and avoiding duplicate identity assignments. The default IoU(overlap) threshold used by Ultralytics is typically around 0.45, and the suppression is applied class-wise during inference.

Together, these filtering steps enhance the precision of person detection and ensure that feature embeddings are based on clean, non-overlapping, and background-reduced image regions.

3.2.4 Visualization and Verification

To support real-time monitoring and offline evaluation, detected bounding boxes are overlaid on the video stream using red rectangles. Each box is annotated with its corresponding confidence score for visibility.

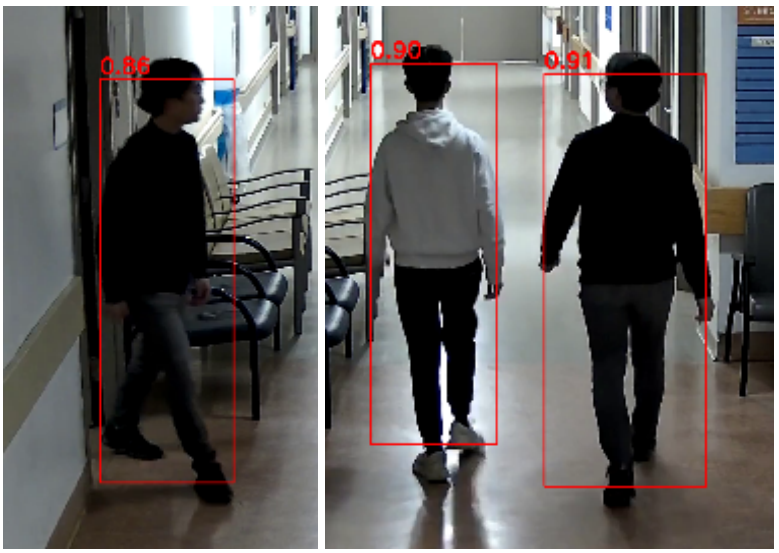


Figure 4. Visualization of YOLOv8 person detection.

This visual feedback is essential during development for:

- Identifying false positives or missed detections.
- Assessing how well the padding and suppression are working.
- Verifying detection timing and alignment across camera feeds.

These visualization tools also help during manual review of Re-ID results, enabling rapid diagnosis of misidentification cases in crowded or visually ambiguous scenarios.

3.3 Re-ID Feature Extractor

Following detection, each person's bounding box is cropped from the video frame and passed to a Re-ID model. The Re-ID module encodes the cropped image into a fixed-length feature embedding vector that captures the person's visual appearance, which can be thought of as the fingerprint of the person's appearance. These embeddings are then used for identity matching across frames and cameras, even if the individual disappears and reappears in different zones. Each feature vector is L2-normalized, enabling efficient comparison via cosine similarity.

This system supports three distinct Re-ID models — PCB, OSNet, and TransReID — each offering different trade-offs between interpretability, robustness, and computational complexity. All three models were integrated in a modular pipeline to allow interchangeable use during experimentation and evaluation. (*Appendix C - Full code*)

3.3.1 Input Preprocessing

Before being passed to a Re-ID model, each detected crop undergoes standard preprocessing:

- The image is resized to the input dimensions required by the target model, (128x384) for PCB, (256x128) for OSNet and TransReID.
- Pixel values are normalized to the range [0, 1].
- The image is converted to channel-first (CHW) format and batched for inference.
- The resulting embedding is L2-normalized before being passed to the global matcher.

3.3.2 PCB-Specific Processing

PCB divides the input image horizontally into six uniform stripes, each corresponding to a part of the human body (e.g., head, torso, legs). In the original PCB model, features are extracted from each horizontal stripe and concatenated to form the final embedding vector. In this project, a custom weighting scheme is used instead,

allowing the model to down-weight facial features and emphasize more consistent appearance cues like shoes or gait. For instance:

- The top stripe, which may contain the face, is excluded entirely.
- Lower body stripes — such as shoes and legs — are given double weight to emphasize consistent appearance cues that remain unchanged even after a patient puts on a hospital gown.

weights = np.array([0, 1, 1, 1, 2, 2])

weights = weights / weights.sum()

The final embedding is a weighted sum of stripe-level features. This setup improves the model's ability to match people across cameras while minimizing reliance on sensitive or variable regions.

However, PCB can struggle when the person is far from the camera. Because PCB rigidly divides each person's image into six equal-height stripes, the resolution of each part can decrease significantly as the full-body crop gets smaller. This is especially problematic in hallway footage, where patients often appear small and distant in the frame. In such cases, each stripe contains fewer pixels, making fine-grained details (e.g., clothing textures or footwear) more blurry or indistinct. Since PCB processes each stripe independently, low-resolution parts contribute noisy or uninformative features, reducing the overall discriminative power of the embedding.

3.3.3 OSNet-Specific Processing

OSNet avoids explicit spatial partitioning. Instead of dividing the body into fixed regions, it extracts features from the entire image crop using a unique architecture based on omni-scale feature learning. Within each residual block (OSBlock), OSNet fuses features from multiple receptive field sizes:

- 1×1 convolutions for fine local textures (e.g., fabric patterns),
- 3×3 and 5×5 convolutions for mid-level cues (e.g., body proportions),
- and larger-scale residual paths for global structure (e.g., silhouette or gait).

This multi-scale fusion enables OSNet to adaptively focus on the most informative spatial scale, depending on the resolution and quality of the input. It proves especially valuable in hallway surveillance settings, where person size varies significantly due to camera placement, and motion blur or compression artifacts can affect appearance consistency.

Unlike PCB, which rigidly splits the person into six equal stripes and requires custom weighting to suppress facial cues and emphasize lower-body appearance, OSNet is used as-is with no model-specific modifications. In this project, the standard pipeline was used:

- The specific OSNet variant used is *os_net_x1_0*, lightweight yet high-performing,
- Each person's crop is resized to the required input size (256x128),
- Passed through the network as a full image,
- And the resulting embedding is L2-normalized for downstream similarity matching.

3.3.4 TransReID-Specific Processing

TransReID is a transformer-based Re-ID model that replaces traditional convolutional backbones with vision transformer (ViT) blocks. Unlike CNNs, which process information hierarchically through local receptive fields, ViTs use global self-attention, enabling the model to aggregate appearance cues across the entire image, even from distant or disjoint parts. This allows TransReID to excel in handling complex visual scenarios such as pose variation, occlusion, viewpoint shifts, and cluttered backgrounds.

In this project, TransReID is loaded using a pretrained configuration:

cfg_path = ".../msmt17/vit_small_ics.yml"

pretrained_path = ".../pretrain/vit_small_ics_cfs_lup.pth"

This configuration uses the ViT-S/16 backbone, which splits the input image into 16×16 patches and applies self-attention over all tokens. The model also incorporates

Instance Centering and Shifting (ICS) — a training-time augmentation strategy that helps the model generalize across camera domains by simulating positional and scale shifts of person crops. This is particularly helpful in hospital footage where camera angles vary and spatial consistency is low.

The model is pretrained on the MSMT17 dataset, one of the largest and most diverse person Re-ID datasets, containing over 4,000 identities captured across 15 different cameras, multiple time slots (day vs. night), and a variety of lighting conditions. MSMT17's diverse camera angles and real-world variability align closely with the conditions of hallway surveillance in hospitals. In particular, it prepares the model to handle: non-overlapping camera views, uncontrolled illumination, and temporal gaps between appearances. This makes MSMT17-pretrained TransReID highly suitable for benchmarking performance in a challenging hospital setup, where patients may appear under different lighting, partially occluded, or after extended delays.

Although TransReID achieves state-of-the-art performance in many Re-ID benchmarks, it is also computationally intensive, both in memory and runtime. As such, it is used in this project only during the evaluation phase to establish an upper bound on embedding accuracy under real-world conditions. It is not planned to be deployed in the final hospital-facing pipeline due to its resource demands, which exceed the capabilities of edge devices like the NVIDIA Jetson.

3.4 Local Tracker with DeepSORT

To stabilize identity assignment within each separated camera stream, this system integrates DeepSORT (Deep Simple Online and Realtime Tracking) as a lightweight multi-object tracking algorithm. While the broader goal of the project is to enable cross-camera person re-identification, local tracking within each camera improves short-term temporal consistency and helps assign a stable ID to each person as they move through the field of view.

Instead of using one global tracker, a separate DeepSORT instance is initialized for each camera, as shown in the code:

self.trackers = [

*DeepSort(max_age=max_age, n_init=n_init, max_iou_distance=max_iou_distance, max_cosine_distance=max_cosine_distance, nn_budget=nn_budget, **embedder=None**, embedder_gpu=embedder_gpu,) for _ in range(num_cams)*

]

This separation avoids potential ID collisions between camera feeds and makes it easier to manage per-camera ID histories, which are later mapped to global IDs using the matcher module.

3.4.1 Tracking Logic and Matching Criteria

DeepSORT links consecutive person detections by combining motion cues (via a Kalman filter) and appearance similarity (via cosine distance between embeddings). However, in this system, the original DeepSORT embedder is bypassed — instead, external Re-ID features extracted from PCB, OSNet, or TransReID are injected directly.

return self.trackers[cam_idx].update_tracks(raw_detections=detections, embeds=features,)

This design ensures that all components of the pipeline — detection, tracking, and matching — use consistent and high-quality Re-ID features, eliminating potential discrepancies introduced by mismatched embeddings.

3.4.2 Key Parameters

Several DeepSORT hyperparameters are carefully tuned for the hospital hallway environment:

- *max_age = 3*: Tracks are removed if no detection is matched for 3 consecutive frames. This short timeout is suitable for real-time environments with relatively stable motion.
- *n_init = 1*: A track becomes “confirmed” after just one detection, reducing startup latency.

- *max_iou_distance = 0.3*: Bounding box overlap threshold for association. Lowering this value helps prevent incorrect matches during occlusions or overlaps.
- *max_cosine_distance = 0.5*: Appearance similarity threshold. Higher thresholds increase recall but risk mismatches.
- *nn_budget = 100*: Number of past appearance features stored per track.

3.4.3 Purpose and Limitations

It's important to note that tracking is not used for cross-camera identity persistence — it only stabilizes short-term trajectories within a single view. As discussed in Section 3.5, global identity assignment is handled separately by the matcher. DeepSORT does not "know" when a person leaves one view and enters another; it simply provides locally consistent track IDs to help reduce flickering IDs during movement.

In future iterations of this system, tracking may be removed entirely, as continuous trajectory linking is not essential to the core task (i.e., estimating time spent in each hospital zone). However, it remains useful for now in minimizing false positives on identity and bounding box in real-time visualizations.

3.5 Global Matcher

In a multi-camera setup where each camera runs an independent DeepSORT tracker, the same individual may be assigned different local track IDs across different views. To address this, a global Re-ID matcher is introduced to consistently associate person embeddings across all camera streams and unify their identities into a shared global ID space.

The *GlobalReIDMatcher* module implements a feature-based identity assignment system that links per-camera tracks into globally consistent identities. This enables cross-camera Re-ID and timeline estimation, even when a patient transitions between hallway zones or temporarily disappears from view.

3.5.1 Design Goals and Challenges

The matcher is designed to address several real-world constraints:

- Camera views are non-overlapping, meaning people reappear in completely different perspectives.
- Trackers are isolated per camera, and cannot share temporal state.
- Embedding features are noisy, especially when the subject is small in frame or partially occluded.
- Global ID assignment must be lightweight and real-time, as the system is designed for edge deployment on devices like NVIDIA Jetson.

To solve these challenges, the matcher builds on three core ideas:

- Confirmed identity memory (*global_memory*)
- Soft candidate memory (*pending*)
- Temporal history buffer (*recent_matches*)

Each person's embedding detected in a frame is processed through this tiered pipeline to determine whether it should be assigned an existing global ID or create a new one.

3.5.2 Matching Logic

The global matching pipeline operates in a priority-based fashion:

1. Match against confirmed memory

If any embeddings in *global_memory* match the input (based on cosine similarity) and are not already used in this frame, they are immediately assigned.

- Matching uses *cdist()* to compute cosine distances.
- A similarity threshold (default 0.35) determines whether a match is strong enough.

- Matching results are checked against *used_ids* to prevent duplicate assignments within the same frame.

2. Temporal memory fallback

If no match is found in the confirmed memory, the matcher checks a recent buffer of past embeddings stored in *recent_matches*. This allows re-identification of individuals who have disappeared for a few seconds (e.g., due to occlusion or movement across cameras).

- Entries older than *temporal_window* (default 5s) are discarded.
- If a recent match is found and not already used, that ID is reused.

3. Pending memory check

If still unmatched, the embedding is compared against the pending list, which tracks candidate IDs that haven't yet been confirmed. A candidate becomes "confirmed" and added to *global_memory* only after it reaches a minimum hit count (default *min_hits=2*). This ensures robustness by avoiding false-positive global ID creation from noisy detections.

4. New ID creation

If the embedding does not match any of the above sources, a new global ID is created, added to pending, and associated with the current feature. The matcher increments an internal counter to ensure global uniqueness.

3.5.3 Memory Update

After global IDs are assigned, *update_memory()* blends new features with existing ones in *global_memory* using momentum averaging, rather than replacing the old feature vector with the new one:

*self.global_memory[i] = (gid, 0.9 * old_feat + 0.1 * new_feat)*

This gradual update method serves several important purposes:

- **Reduces noise and one-off errors**: Sometimes, a new detection might be affected by lighting changes, occlusions, or motion blur. If we replaced the stored embedding with this noisy feature, it could degrade identity consistency. Blending instead ensures that short-term noise does not dominate the identity representation.

- **Smooths the identity over time**: People don't look exactly the same from frame to frame — they might rotate slightly, walk differently, or shift position. Momentum averaging helps capture a more stable and time-averaged representation of the person, improving robustness across camera views.

- **Prevents catastrophic drift**: Simply overwriting with each new feature would make the system overly reactive. A few outlier frames (e.g., poor crops, extreme occlusion) could pull the embedding far from the person's true appearance. By weighting older features more heavily (90%), the system anchors the identity while still slowly adapting.

- **Maintains Re-ID accuracy over long durations**: In long-term tracking across multiple rooms, a person's appearance may change subtly, for example, due to lighting shifts or small movements. Momentum averaging enables the system to gradually adapt, preserving performance over time.

This technique is a simple but effective way to balance stability and adaptability, which is especially important in noisy, real-world environments like hospital hallways.

3.5.4 Identity Mapping

Each camera maintains a *local_to_global* dictionary externally to map DeepSORT-assigned *track_ids* to global IDs. This ensures that the correct label is

drawn in the visualization pipeline, even if the same person appears in different views or switches between camera zones.

3.5.5 Discussion and Benefits

This tiered matcher:

- Combines short-term consistency (via tracker) with long-term Re-ID memory (via embeddings),
- Works without any explicit calibration or camera synchronization,
- Tolerates brief occlusions and fragmented trajectories,
- Can be extended with spatial cues or domain-specific constraints in the future.

In short, the *GlobalReIDMatcher* bridges isolated detections into a unified identity timeline — a crucial step for understanding how long patients spend in specific hospital areas.

4. Evaluations

4.1 Evaluation Setup

To assess the system's effectiveness under realistic hospital conditions, evaluations are conducted using a set of manually recorded video clips collected at St. Michael's Hospital, Toronto. Three ceiling-mounted Reolink IP cameras were used to capture the hallway area in front of the CT scan room and changing room, as shown in *Appendix B*. Each evaluation sequence contained 1–4 people walking independently or in overlapping groups through the monitored area. The goal of the evaluation was to test whether the system could consistently assign the same global identity to a person across multiple camera angles, even with variations in lighting, camera perspective, appearance, and motion blur.

The tests were conducted on a local development machine, as well as a Jetson-compatible deployment version using OSNet for real-time inference. YOLOv8

was used for person detection, while different Re-ID models (PCB, OSNet, TransReID) were evaluated separately on the same video sequences for comparison.

There are three 7-minute-long videos prepared, captured by three cameras, covering a variety of conditions:

- Single-person tracking across all three cameras.
- Single person appears and disappears from the camera views.
- Two individuals wearing similar clothing.
- Two or more individuals appear simultaneously in the same hallway.
- Partial occlusions due to people crossing paths.
- Low-resolution appearances when subjects are far from the camera.
- Appearance variation, including cases where patients wore hospital gowns.

These scenarios were chosen to reflect common challenges faced in hallway Re-ID pipelines and to stress-test identity preservation across views.

4.2 Quantitative Evaluation (Not Conducted)

This project focuses on building a real-world hospital hallway Re-ID pipeline, rather than optimizing for benchmark accuracy. As such, quantitative evaluation was not conducted, primarily due to the absence of ground-truth identity labels for the collected video data. Annotating person IDs for thousands of frames across multiple camera views in a hospital setting is both time-consuming and error-prone, especially when faces are obscured or appearance changes (e.g., gown usage) occur.

Instead, the system was evaluated qualitatively through carefully designed test scenarios, including single- and multi-person tracking, partial occlusions, low-resolution appearances, and appearance variation. These qualitative observations revealed practical strengths and limitations of different Re-ID models, providing valuable insights into their deployment performance under real-world constraints.

While standard Re-ID metrics such as mean Average Precision (mAP) or Cumulative Matching Characteristic (CMC) curves were not computed, future work may

incorporate quantitative benchmarking using synthetic or manually labeled hospital footage to complement the current qualitative results.

4.3 Qualitative Evaluations

To evaluate the system's ability to assign consistent global IDs across different camera views and scenarios, qualitative testing is conducted across the three recorded evaluation sequences. These clips included a variety of real-world hallway situations designed to test identity preservation under difficult conditions. The Re-ID models evaluated include PCB, OSNet, and TransReID, each tested independently using the same detections and input frames.

Single-person tracking across all cameras

All three models successfully preserved identity when tracking a single individual moving through all three cameras. The assigned global ID remained stable. However, PCB became unstable when the person moved further from the camera, likely due to lower resolution in its part-based feature stripes. TransReID took longer to stabilize the identity, but once locked in, it remained reliable. OSNet performed the most consistently, quickly stabilizing and maintaining the correct global ID throughout the person's entire path.

Figure 5. Single-person tracking across hallway cameras using OSNet.

All three Re-ID models (PCB, OSNet, and TransReID) maintained a consistent global identity for the individual.

Single person appears and disappears from view

In this test, an individual walked into and out of each camera's field of view. All three models handled this case effectively, with global IDs preserved before and after reappearance, confirming the system's ability to re-identify individuals even with temporal gaps in visibility.

Figure 6. Re-identification of a person who disappears and later reappears in the hallway.

Output from OSNet, PCB and TransReID also preserved the identity in this scenario

Partial occlusion (crossing paths)

All three models were able to recover from partial occlusions, such as people walking in front of each other, by waiting for more frames to resolve ambiguity. Once re-identified in the following frames, the correct global ID was restored, showing the benefit of frame-level independence in the Re-ID pipeline.

Two individuals wearing similar clothing

This scenario tested the system's ability to distinguish between individuals with near-identical appearances (similar dark sweaters and pants). After tuning the similarity threshold, OSNet was able to consistently differentiate between them, while PCB and TransReID struggled, often assigning the same global ID to both individuals. This shows OSNet's greater sensitivity to subtle visual cues and better performance in appearance-based discrimination.

Figure 7. OSNet successfully distinguishes two individuals with similar clothing across different camera views. Despite similar upper-body appearance, the model assigns consistent and distinct global IDs (ID 1 and ID 7)

Figure 8. Failure case: Both PCB and TransReID mistakenly assign the same global ID to two different individuals with similar clothing.

Multiple individuals in the same frame

When two or more individuals appeared simultaneously and moved side by side through the hallway, only OSNet was able to handle this case robustly. Although TransReID is theoretically more powerful, it did not perform as well under this

scenario, likely due to the need for further fine-tuning on hospital footage. PCB performed poorly, frequently assigning the same ID to overlapping individuals, likely due to its rigid part-based structure and poor spatial resolution in crowded scenes.

Figure 9. Example of OSNet successfully distinguishing multiple individuals consistently in the same frame.

Low-resolution appearances (far distance)

When individuals were far from the camera and appeared small in the frame, OSNet maintained ID stability, while PCB struggled significantly, likely due to insufficient pixel information in individual stripes. TransReID also performed worse than OSNet in these cases, indicating limitations in its ability to handle low-resolution inputs without additional fine-tuning.

Figure 10. Example output from PCB with low-resolution appearance from far-field views.

Appearance changes (e.g., hospital gowns)

Currently, none of the models are able to reliably maintain identity when patients change into hospital gowns, which significantly alter visual features like clothing and body outline. This motivates future work toward entry/exit-based identification, where cameras can be placed closer to the subject (near doorways), and where facial data could be optionally used to enhance identity consistency.

Processing Speed

In addition to accuracy and ID stability, processing speed is a critical consideration for real-time deployment in hospital environments. Based on runtime observations during evaluation, each Re-ID model demonstrated noticeably different performance characteristics.

When processing video frames containing multiple people (4–5 individuals appearing simultaneously), TransReID exhibited a visible slowdown compared to both OSNet and PCB. This is expected, as TransReID uses a transformer-based architecture with self-attention layers, which are computationally intensive and scale poorly with image size and the number of detections. Notably, even during inference, TransReID remains computationally expensive, making it challenging to deploy on resource-constrained devices like the NVIDIA Jetson.

In contrast, OSNet and PCB rely on lightweight convolutional operations, which are significantly more efficient. OSNet maintained smooth frame rates even in crowded scenes and was more consistent than PCB, which, while fastest, lacked robustness in complex tracking scenarios.

4.4 Limitations and Future Work

While the current system successfully demonstrates hallway-based person Re-ID using multiple cameras, several limitations were observed during testing. These limitations point to opportunities for refining the approach and guiding future development.

Limited use of temporal and spatial priors:

While the system primarily relies on appearance-based features (via PCB, OSNet, or TransReID) for identity assignment, a lightweight temporal memory was implemented via the *recent_matches* buffer in the global matcher. This allowed the system to favor recent IDs within a short time window, helping to reduce identity switches. However, this temporal logic is relatively shallow and lacks awareness of spatial transitions between cameras. When visually similar individuals appear in different camera views simultaneously, identity collisions can still occur.

Future improvements could involve more structured spatio-temporal modeling, such as defining camera transition zones or using entry/exit timing constraints. These additions would make the system more robust in high-traffic areas or overlapping view scenarios.

Performance drop with low-resolution and partial visibility:

When subjects are far from the camera or partially occluded, PCB and TransReID embeddings become unstable. This is particularly problematic in long hallways where people appear small or blurred. One possible way to fix this is to place cameras closer to people, using higher-resolution cameras, or choose models like OSNet that work better when people appear small or blurry.

Identity inconsistency with appearance changes:

The system struggles when a person changes into a hospital gown mid-process. To address this, PCB was tested as a part-based Re-ID model, with a custom weighting scheme that emphasized lower body features like footwear while suppressing facial features. However, in practice, PCB failed to deliver reliable identity preservation, especially when subjects were far from the camera or partially occluded. The part-level features became too blurry to distinguish effectively. Future work could explore more specialized footwear detectors or gait-based Re-ID to

increase robustness to clothing variation.

Unnecessary use of continuous tracking:

Although the system includes DeepSORT to provide smooth, real-time identity tracking within each camera view, testing revealed that continuous tracking is not essential for meeting the project's goals. The main requirement is to detect when a patient enters and exits key hospital areas, such as the changing room or CT room, rather than maintaining a continuous trajectory across all hallway frames.

Moreover, continuous tracking increases the number of frames processed per person, which means more frequent updates to the global feature memory. Even with momentum averaging (90% old feature + 10% new feature), repeated updates over noisy or low-quality frames can degrade the identity embedding over time. This could lead to ID drift or even incorrect global assignments.

In contrast, an entry/exit-based approach would evaluate only a few carefully selected frames per person, ideally when the person is fully visible and close to the camera. This minimizes the risk of noisy feature updates, reduces computational load, and may result in more stable and accurate Re-ID performance. Therefore, future iterations of the system could eliminate continuous tracking in favor of event-based identity assignment.

No facial features were used, but could be optionally included:

The project avoided using facial recognition for privacy reasons. However, in close-range setups (door-mounted cameras), facial embeddings could serve as a powerful fallback for Re-ID, especially in scenarios where clothing or body shape offers limited information. A hybrid pipeline that selectively includes facial data when available may provide the best trade-off between accuracy and privacy.

5 Conclusion

This thesis presented a hallway-based multi-camera person re-identification system aimed at tracking patients through key steps in a medical imaging workflow, such as entering and exiting changing rooms or CT scan rooms. By combining person detection (YOLOv8), per-camera tracking (DeepSORT), and robust visual embedding models (PCB, OSNet, TransReID), the system attempts to assign consistent global IDs to each patient across multiple cameras and appearance changes.

The system was designed with privacy, deployability, and hospital constraints in mind. Facial recognition was avoided, and all processing was performed locally. OSNet was selected for final deployment due to its balance between speed, accuracy, and compatibility with Jetson devices. TransReID served as a high-performance benchmark, while PCB was used to explore part-based identity reasoning.

Evaluation on real hospital-collected video data showed that the system works reliably in simple cases, such as single-person movement across cameras. However, several challenges remain, including failures in distinguishing people with similar clothing, handling blurry views, and appearance changes like hospital gowns. Additionally, the use of continuous tracking (via DeepSORT) was found to introduce complexity without always improving performance.

Looking forward, the project proposes a streamlined Re-ID pipeline that eliminates continuous tracking and instead focuses on detecting patient entry and exit at specific hospital zones. Such a redesign is more aligned with the system's core goal: estimating how long patients spend at each step, rather than tracking every movement. Future improvements could also include better spatial modeling between cameras or integrating facial or biometric cues in a privacy-conscious way.

References

[1] Y. Y. Cheung, E. M. Goodman, and T. O. Osunkoya, “No More Waits and Delays: Streamlining Workflow to Decrease Patient Time of Stay for Image-guided Musculoskeletal Procedures,” Radiographics: A Review Publication of the Radiological Society of North America, Inc, vol. 36, no. 3, pp. 856–871, 2016, doi: 10.1148/rg.2016150174.

[2] L. Zhang, A. Hefke, J. Figiel, U. Schwarz, M. Rominger, and K. J. Klose, “Enhancing SameDay Access to Magnetic Resonance Imaging,” Journal of the American College of Radiology: JACR, vol. 8, no. 9, pp. 649–656, Sep. 2011, doi: 10.1016/j.jacr.2011.04.001.

[3] C. J. Roth, D. T. Boll, L. K. Wall, and E. M. Merkle, “Evaluation of MRI Acquisition Workflow with Lean Six Sigma Method: Case Study of Liver and Knee Examinations,” AJR. American journal of roentgenology, vol. 195, no. 2, pp. W150–156, Aug. 2010, doi: 10.2214/ AJR.09.3678.

[4] G. Revesz, F. J. Shea, and M. C. Ziskin, “Patient Flow and Utilization of Resources in a Diagnostic Radiology Department,” Radiology, vol. 104, no. 1, pp. 21–26, Jul. 1972, doi: 10.1148/104.1.21.

[5] A. Brown et al., "CT YOLO-AR: Real-time Medical Imaging Process Analytics," St. Michael’s Hospital, University of Toronto, 2024.

[6] L. Zheng, Y. Yang, and A. G. Hauptmann, "Person Re-identification: Past, Present and Future," *arXiv preprint arXiv:1610.02984*, 2016.

[7] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, “Scalable Person Re-identification: A Benchmark,” *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015.

[8] Y. Wei et al., “GLAD: Global-Local-Alignment Descriptor for Pedestrian Retrieval,” *ACM Multimedia*, 2017.

[9] J. Redmon and A. Farhadi, “YOLOv3: An Incremental Improvement,” *arXiv preprint arXiv:1804.02767*, 2018.

[10] N. Wojke, A. Bewley, and D. Paulus, “Simple Online and Realtime Tracking with a Deep Association Metric,” *2017 IEEE International Conference on Image Processing (ICIP)*, 2017.

[11] Y. Sun, Q. Xu, Y. Li, C. Zhang, Y. Zhang, and J. Sun, “Beyond Part Models: Person Retrieval with Refined Part Pooling,” *ECCV*, 2018.

[12] K. Zhou, Y. Yang, A. Cavallaro, and T. Xiang, “Omni-Scale Feature Learning for Person Re-Identification,” *ICCV*, 2019.

[13] C. He, Y. Luo, P. Wang, F. Li, H. Li, and F. Wu, “TransReID: Transformer-based Object Re-Identification,” *ICCV*, 2021.

Appendices

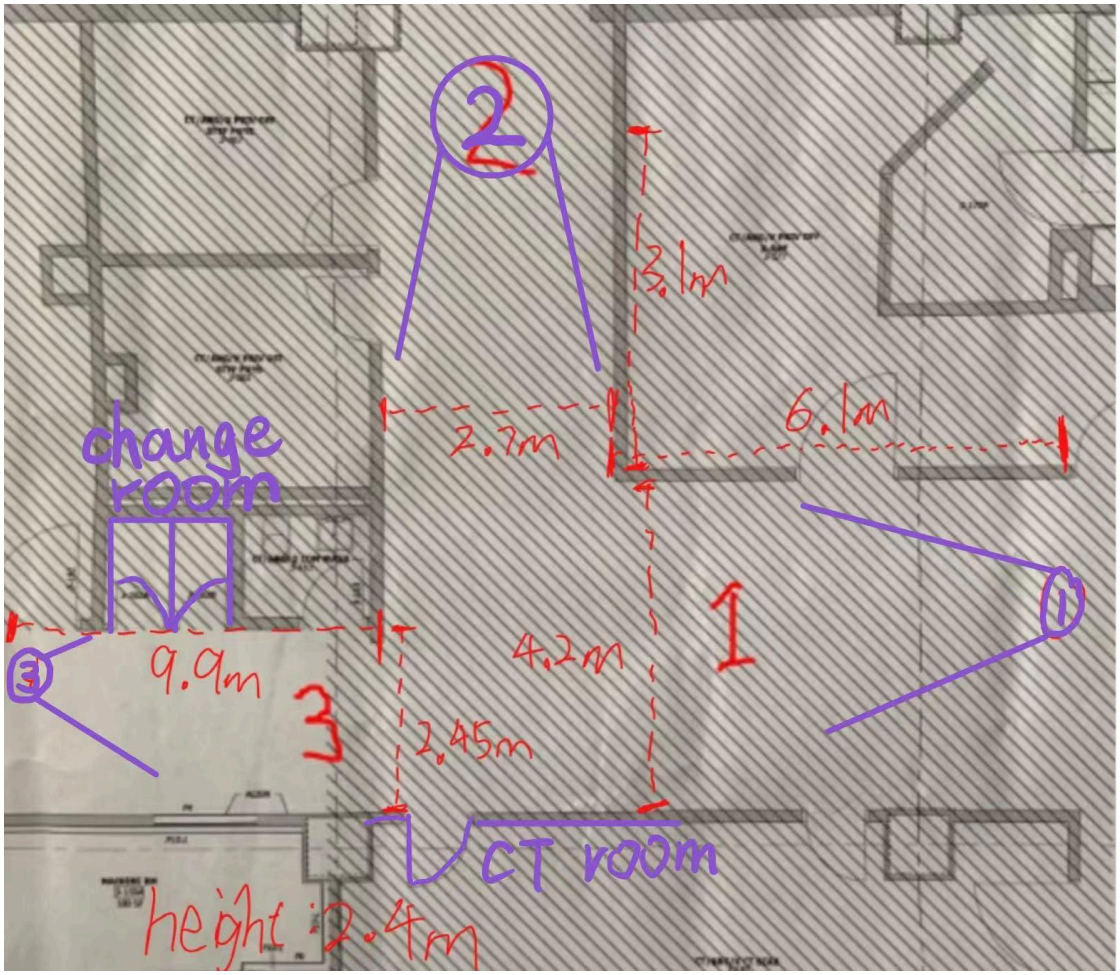
Appendix A - Perceptual Scales in OSNet

Scale	Receptive Field (Filter Size)	What It Captures	Examples Visual Cues
Local	1x1	Fine-grained, highly localized details	Texture, color patches, accessories
Mid	3x3	Structural and shape-level information	Clothing shape, body contours
Global	5x5	Coarse features over larger spatial context	Silhouette, gait, overall body form

The concept of *multiple perceptual scales* in OSNet refers to the network's ability to analyze an image at different levels of spatial detail, simultaneously. This is achieved using the OSBlock, which integrates features from convolutional filters with different receptive field sizes. Each receptive field size captures information at a different "scale" of the image:

- 1×1 filters capture very local, fine-grained details, such as texture or color patterns in a small patch.
- 3×3 filters focus on mid-level features, such as shapes, edges, or clothing patterns.
- 5×5 filters and larger ones are better at extracting coarse, high-level features, like body posture or silhouette.

Appendix B - Camera Setup Layout



Appendix C - Complete Code

Full Code at: https://github.com/anguskk109/MultiCamReID

Detector - Choose target class = person and apply shrink padding

```
for box in results.boxes:
    x1, y1, x2, y2 = map(int, box.xyxy[0])
    conf = box.conf[0].item()
    cls = int(box.cls[0])

    if cls == 0 and conf > 0.6:  # Only person, confidence threshold
        # Shrink the box slightly to cut background
        pad = 0.05
        w = x2 - x1
        h = y2 - y1
        x1_new = max(int(x1 + pad * w), 0)
        y1_new = max(int(y1 + pad * h), 0)
        x2_new = min(int(x2 - pad * w), width - 1)
        y2_new = min(int(y2 - pad * h), height - 1)
```

Matcher - Temporal Filtering and Momentum Averaging Feature Update Rule

```
def _temporal_match(self, feature, now, used_ids):
    if not self.recent_matches:
        return None
    recent_feats = np.array([f for _, f, t in self.recent_matches if now - t < self.temporal_window])
    if recent_feats.size == 0:
        return None
    distances = cdist([feature], recent_feats, metric="cosine")[0]
    best_idx = np.argmin(distances)
    best_dist = distances[best_idx]
    if best_dist < self.similarity_threshold:
        gid_candidates = [gid for gid, _, t in self.recent_matches if now - t < self.temporal_window]
        gid = gid_candidates[best_idx] if gid_candidates[best_idx] not in used_ids else None
        return gid
    return None

def _update_recent(self, gid, feature, now):
    self.recent_matches.append((gid, feature, now))
    # Clean old entries
    self.recent_matches = [(g, f, t) for (g, f, t) in self.recent_matches if now - t < self.temporal_window]

def update_memory(self, ids, features):
    # Ensure newest feature is updated in memory
    for gid, feat in zip(ids, features):
        for i, (stored_id, old_feat) in enumerate(self.global_memory):
            if stored_id == gid:
                self.global_memory[i] = (gid, 0.9 * old_feat + 0.1 * feat)
                break
```

Tracker - Initialize tracker for each camera

```python
from deep_sort_realtime.deepsort_tracker import DeepSort

class MultiCameraTracker:
    def __init__(
        self,
        num_cams,
        max_age=3,
        n_init=1,
        max_iou_distance=0.3,
        max_cosine_distance=0.5,
        nn_budget=100,
        embedder=None,
        embedder_gpu=True
    ):
        """
        Initializes a separate DeepSORT tracker for each camera.
        """
        self.trackers = [
            DeepSort(
                max_age=max_age,
                n_init=n_init,
                max_iou_distance=max_iou_distance,
                max_cosine_distance=max_cosine_distance,
                nn_budget=nn_budget,
                embedder=embedder,
                embedder_gpu=embedder_gpu,
            )
            for _ in range(num_cams)
        ]
```

Main - General Workflow: Detect - Feature Extraction - Match - Track

```
detections = detector.detect(frame, visualize=True)
if detections:
    features = [extract_osnet_features(reid_model, frame[y1:y2, x1:x2])
                for (x1, y1, x2, y2, _) in detections]
    formatted = [[[x1, y1, x2, y2], conf, "person"] for (x1, y1, x2, y2, conf) in detections]

    # Step 1: Match features globally
    global_ids = global_matcher.match(features)
    global_matcher.update_memory(global_ids, features)

    # Step 2: Update tracker
    tracks = tracker.update(i, formatted, features, frame)

     # Step 3: Map detection box to GID
    detection_to_gid = {
        tuple([x1, y1, x2, y2]): gid
        for ((x1, y1, x2, y2, _), gid) in zip(detections, global_ids)
    }

    # Step 4: Map local track_id to global_id using original detection box
    for track in tracks:
        if track.original_ltwh is not None:
            x, y, w, h = track.original_ltwh
            x1, y1 = int(x), int(y)
            x2, y2 = int(x + w), int(y + h)
            box = (x, y, w, h)

            gid = detection_to_gid.get(box)
            if gid is not None:
                global_matcher.local_to_global[i][track.track_id] = gid


else:
    tracks = []


# Step 5: Draw tracks with global ID (fallback-aware)
for track in tracks:
    if track.is_confirmed() and track.time_since_update == 0:
        gid = global_matcher.local_to_global[i].get(track.track_id, f"?")
        draw_tracks(frame, [track], label=f"ID {gid}")
```