

# Multi-Camera Vehicle Tracking Based on the ELECTRICITY Framework

Yuxin Chen  
University of Arkansas  
Fayetteville, AR  
yuxinc@uark.edu

Luke Waind  
University of Arkansas  
Fayetteville, AR  
ldwaind@uark.edu

Samanwita Das  
University of Arkansas  
Fayetteville, AR  
sdas@uark.edu

## Abstract

*Multi-Camera Multi-Target (MCMT) tracking is a fundamental task in intelligent transportation, requiring the consistent identification of vehicles across spatially disjoint camera views. In this project, we explore the challenges of the MCMT pipeline, specifically focusing on the critical role of vehicle Re-Identification (ReID). We utilize the **ELECTRICITY** model, an efficient tracklet-association architecture, to study tracking performance across domains. Our primary contribution is the fine-tuning of the **ELECTRICITY** ReID component on two distinct datasets: the synthetic **Synthehicle** dataset and the real-world **CityFlow** benchmark. Additionally, we utilize the CARLA simulator to generate simple video samples, gaining insight into the synthetic data generation process. We evaluate our fine-tuned models using IDF1, ID Precision (IDP), and ID Recall (IDR), providing a comparative analysis of tracking performance in simulated versus real-world environments.*

## 1. Introduction

As urban environments become increasingly monitored by surveillance networks, the ability to track vehicles across multiple non-overlapping cameras—known as Multi-Camera Multi-Target (MCMT) tracking—has become a cornerstone of Intelligent Transportation Systems (ITS). Unlike Single-Camera Tracking (SCT), which focuses on continuity within a single view, MCMT must resolve identity consistency across significant spatiotemporal gaps. This presents a unique challenge: the appearance of a vehicle can change drastically due to lighting, pose, and occlusion differences between cameras. MCMT proves to be a much more challenging task than SCT due to the significant spatiotemporal gaps and varying viewpoints between each camera’s perception of vehicles.

Despite much progress and great strides taken in the area of deep learning-based tracking, a critical bottleneck remains: the scarcity of large-scale, accurately annotated real-world datasets. Manual annotation of multi-camera footage

is prohibitively expensive. To address this, the “Synthehicle” project [6] proposes leveraging synthetic data generation to cheaply generate hours of realistic data with flawless, extensively annotated data.

This report documents our reproduction of the Synthehicle experiments. We implemented the **ELECTRICITY** model [10] as our primary tracker to verify its reported performance on synthetic benchmarks. Furthermore, we address the challenge of domain adaptation by testing the synthetic-trained model on the real-world **CityFlow** dataset [12], providing insights into the feasibility of transferring models trained in virtual environments (CARLA [3]) to physical urban surveillance networks.

### 1.1. Motivation

The primary bottleneck in advancing MCMT research is the reliance on large-scale, annotated datasets. Collecting real-world data is labor-intensive and expensive due to the privacy constraints and the difficulty of manually synchronizing identities across cameras. Synthetic data generation, using simulators like CARLA, offers a promising alternative by providing perfectly annotated data with controllable parameters (weather, density, viewpoint). However, models trained on synthetic data often suffer from the “domain gap”, a degradation in performance when applied to real-world footage due to differences in texture and physics.

### 1.2. Project Objectives

This project aims to dissect the MCMT pipeline and evaluate the efficacy of current deep learning methods in both synthetic and real domains. Specifically, our contributions are:

- **Pipeline Implementation:** We implemented the full MCMT pipeline using the **ELECTRICITY** model [10], focusing on its graph-based association capabilities.
- **ReID Fine-Tuning:** We fine-tuned the model’s Re-Identification (ReID) component separately on the synthetic **Synthehicle** dataset and the real-world **CityFlow** dataset [12].
- **Synthetic Generation Exploration:** We utilized the

CARLA simulator [3] to generate simple video sequences, allowing us to understand the mechanisms behind synthetic data creation, such as camera synchronization and asset spawning.

- **Evaluation:** We assessed the model’s tracking robustness using standard metrics: IDF1, ID Precision (IDP), and ID Recall (IDR). This comparison highlights the performance disparities between identifying “perfect” synthetic vehicles and noisy real-world targets.

## 2. Background

The Multi-Camera Multi-Target (MCMT) tracking pipeline is complex, typically decomposing into two hierarchical stages: Single-Camera Tracking (SCT) and Inter-Camera Tracking (ICT).

### 2.1. Single-Camera Tracking (SCT)

In the first stage, vehicles are detected and tracked within the field of view of a single camera to form “tracklets.”

- **Detection:** Modern pipelines utilize real-time object detectors such as the **YOLO** (You Only Look Once) family [11] to localize vehicles in each frame.
- **Motion Estimation:** To associate detections across frames, algorithms like **SORT** (Simple Online and Realtime Tracking) [1] use Kalman filters to predict object trajectories.
- **Appearance Integration:** To handle occlusions where motion prediction fails, **DeepSORT** [14] extends SORT by integrating deep appearance descriptors, reducing identity switches when targets overlap.

### 2.2. Inter-Camera Tracking (ICT) and Re-Identification

The ICT stage associates these local tracklets across the global camera network. This relies heavily on **Vehicle Re-Identification (ReID)**, which aims to match images of the same vehicle captured from different viewpoints.

- **Feature Extraction:** Deep Convolutional Neural Networks (CNNs), particularly **ResNet-101** [4], serves as the standard backbone for extracting robust feature embeddings.
- **Loss Functions:** To learn discriminative features, models are typically trained using **Triplet Loss** [5], which ensures that embeddings of the same vehicle are closer in vector space than those of different vehicles.

### 2.3. The ELECTRICITY Model

Our project utilizes the **ELECTRICITY** model for the ICT stage. Unlike traditional methods that treat association as a heavy graph-cut problem, **ELECTRICITY** formulates it as an efficient energy minimization task. It generates a “tracklet graph” where nodes are single-camera tracklets

and edges represent the similarity cost derived from both spatiotemporal constraints and ReID feature distances.

## 3. Related Work

### 3.1. Multi-Camera and Single-Camera Tracking

Tracking logic has evolved from simple motion-based heuristics to complex deep learning frameworks. Early SCT methods relied on Kalman filters (e.g., SORT [1]), while modern approaches integrate appearance features (e.g., DeepSORT [14]) to handle occlusions. In the multi-camera domain (MCMT), the challenge expands to global optimization. Recent works often model the camera network as a graph, where tracklets are nodes and edges represent similarity scores. The **ELECTRICITY** model exemplifies this by treating tracking as an energy minimization problem, allowing for highly efficient online association compared to computationally expensive offline graph-cut methods [7].

### 3.2. Cross-Camera Tracking and Re-Identification (ReID)

The core of MCMT is ReID—matching vehicle identities across views. This is challenging due to high intra-class similarity (e.g., many white sedans look alike). Baseline methods use Triplet Loss or Contrastive Loss to learn discriminative embeddings. More advanced techniques, such as **VehicleX** [15], focus on learning view-invariant features. Recent research has also introduced attention mechanisms to focus on unique vehicle details (stickers, ornaments) rather than general shape, which is critical for distinguishing visually similar targets in datasets like CityFlow [12].

### 3.3. Synthetic Data and Sim-to-Real Transfer

To overcome data scarcity, the field has turned to simulation.

- **Advantages:** Simulators like **CARLA** [3] allow researchers to generate infinite data without manual annotation. They provide full control over environmental variables (lighting, rain, camera angle). The **Synthevehicle** dataset [6] leverages this to create a massive, consistent virtual city benchmark.
- **Challenges (Domain Gap):** Despite these advantages, a gap exists between simulation and reality. Features learned from rendered textures often fail to generalize to real camera noise. Recent work in Domain Adaptation [8] attempts to bridge this by aligning feature distributions between synthetic and real domains. Our work empirically observes this gap by comparing ReID performance on the clean Synthevehicle baseline versus the noisy CityFlow environment.

## 4. Electricity model

### 4.1. Vehicle Detection

The ELECTRICITY model uses Mask R-CNN as the vehicle detection model in the detection phase, performing target detection and instance segmentation on each frame. To accomplish multi-camera, multi-target tracking, the first step is to identify targets in video files. Video files consist of multiple consecutive image files. Therefore, identifying targets in a video file is transformed into identifying targets in each frame of the video file.

Mask R-CNN uses a deep convolutional neural network as the feature extraction backbone and incorporates the RoI Align mechanism to generate candidate target regions from the image. Its multi-head structure allows it to simultaneously output target category information, confidence scores, bounding box coordinates, and pixel-level segmentation masks.

The ELECTRICITY model focuses on three target categories: Car, Bus, and Truck, treating them all as vehicle categories for subsequent processing. Because different categories have similar appearances, performing non-maximum suppression (NMS) only within a single category can easily lead to duplicate detection problems. To address this, the model further incorporates Inter-class Non-maximum Suppression (Inter-class NMS) on top of conventional NMS. This NMS performs cross-class filtering based on the Intersection over Union (IoU) and confidence scores between detection boxes, effectively eliminating duplicate detections of the same vehicle. The vehicle detection results obtained by the ELECTRICITY model at this stage will provide input for subsequent single-camera multi-object tracking and cross-camera target association.

### 4.2. Multi-target Single-camera Tracking

In ELECTRICITY model, multi-target tracking within a single camera uses the idea of "tracking-by-detection." This means first detecting vehicles in each frame, then associating the detection results from adjacent frames to form continuous single-camera tracks.

The ELECTRICITY model uses the DeepSORT algorithm as the online multi-target tracking method during the single-camera tracking phase. For each vehicle target, the system uses a Kalman filter to build its motion state model and predicts the target's position at the next moment under the assumption of uniform motion. Since traffic videos typically have a high frame rate (30 FPS), the displacement of vehicles between adjacent frames is small, thus this motion prediction model exhibits good stability and reliability.

During the target association process, DeepSORT simultaneously incorporates:

- Motion information: the spatial position of the target predicted by the Kalman filter.

- Appearance information: the appearance similarity of the targets described by depth features.

Unlike traditional DeepSORT methods, ELECTRICITY does not train a separate pedestrian or vehicle appearance feature extraction network. Instead, it directly reuses the deep features of the Region of Interest (RoI) in the Mask R-CNN detection network as appearance description vectors, for reducing computational complexity.

The ELECTRICITY model first uses appearance features and spatial constraints to match confirmed trajectories with the detection results of the current frame. For targets that still do not match, a second matching is performed using the Intersection over Union (IoU) of bounding boxes. Finally, detection results that still do not match are initialized as new trajectories. Through this single-camera multi-target tracking process, the model can associate vehicle detection results frame by frame into continuous trajectories, providing input for cross-camera vehicle re-identification and multi-camera target association.

### 4.3. Vehicle Re-identification

After obtaining single-camera tracklets, the model applies vehicle re-identification (ReID) to match the same vehicle across different cameras. The ReID task focuses on learning discriminative appearance features for vehicle matching.

#### 4.3.1. Training Stage and Aggregation Loss

During training, images from different vehicle identities are sampled to form a batch. Each image is sent into a deep neural network  $f(\cdot)$  to extract an appearance feature vector. The ReID network is optimized using an aggregation loss, which combines triplet loss and cross-entropy loss:

$$L_{\text{agg}} = \alpha L_{\text{tr}} + \beta L_{\text{xe}} \quad (1)$$

where  $L_{\text{tr}}$  is the triplet loss,  $L_{\text{xe}}$  is the cross-entropy loss, and  $\alpha, \beta$  are weighting factors.

#### 4.3.2. Triplet Loss with Hard Mining

Triplet loss is used to enforce that features of the same vehicle are close, while features of different vehicles are far apart. With hard mining, the triplet loss is defined as:

$$L_{\text{tr}} = \max(0, L_{\text{hp}} + L_{\text{hn}} + \lambda) \quad (2)$$

The hard positive samples are the most different images of the same vehicle, and hard negative samples are the most similar images from different vehicles.

The hard positive loss is:

$$L_{\text{hp}} = \sum_{i=1}^V \sum_{j=1}^{B_i} \left( \max_{k=1}^{B_i} D \left( f(I_i^j), f(I_i^k) \right) \right) \quad (3)$$

The hard negative loss is:

$$L_{hn} = - \sum_{i=1}^V \sum_{j=1}^{B_i} \left( \min_{\substack{p=1..V, q=1..B_p \\ p \neq i}} D(f(I_i^j), f(I_p^q)) \right) \quad (4)$$

Where  $B_i$  represents the number of images belonging to vehicle  $i$ , and  $I_i^j$  represents the  $j$ -th image of vehicle  $i$ . The function  $D(\cdot)$  denotes the distance function in the feature space.  $L_{hp}$  is the hard positive loss, which measures the largest feature distance between images of the same vehicle.  $L_{hn}$  is the hard negative loss, which measures the smallest feature distance between images of different vehicles. Through this loss design, the model is trained to reduce the distance between the hardest positive samples and increase the distance between the hardest negative samples.

#### 4.4. Multi-target Cross-camera Tracking

After completing multi-target tracking within a single camera and vehicle re-identification across cameras, the model needs to uniformly associate vehicle tracklets from different cameras to assign a unique global ID to the same vehicle.

Input of this component consists of the single-camera tracks generated from each camera and vehicle appearance features extracted by the ReID component. The model first calculates the distance between appearance features of each pair of tracks from different cameras, combined with the location relationship between cameras. Since vehicles in real-world scenarios can only pass adjacent cameras sequentially, the model only associates logically adjacent cameras.

In the cross-camera association process, the model judges based on the following rules:

1. If the distance between the appearance features of two tracks is smaller than a preset threshold, they are determined to belong to the same vehicle.
2. If the distance between the appearance features of a track and all other tracks from other cameras is larger than the threshold, the track is regarded as an unmatchable isolated track.
3. Cross-camera association is performed step by step according to the chronological order and camera order to ensure the temporal consistency of vehicle movement.

Through this strategy, the model can unify the tracklets of the same vehicle from different cameras with a single global id.

## 5. Experiments

Our experimentation served two purposes. For our first experiment, we input the available Syntheicle dataset into a pretrained model offered by the ELECTRICITY team to get preliminary MTMCT results in the same way that the authors of the Syntheicle paper did. Following that step, we trained two instances of the ELECTRICITY model, one

using Syntheicle data and the other using CityFlow. With these two instances, we performed evaluation on the ReID component of the ELECTRICITY model to compare the results from synthetic and real-world datasets.

### 5.1. Evaluation Metric and Datasets

**Evaluation Metric.** IDF1 is the official evaluation metric used on the leaderboard for multi-object and multi-camera tracking. Let IDTP denote the number of true positive identities, IDFP denote the number of false positive identities, and IDFN denote the number of false negative identities. The IDF1 score is defined as:

$$\text{IDF1} = \frac{2\text{IDTP}}{2\text{IDTP} + \text{IDFP} + \text{IDFN}} \quad (5)$$

#### 5.1.1. Syntheicle Dataset

We used the pre-generated synthetic dataset provided by the Synthetic official GitHub repository. The dataset is organized into multiple virtual towns. Each town is captured under four different time and weather conditions: day, night, dawn, and rain.

#### 5.1.2. CityFlow Dataset

We used the AI City 2022 Challenge dataset (Track 1) for both training and evaluation. For training, we adopted Scene S01 from the training set, which contains data from 5 different cameras. For evaluation, we used the validation dataset, which consists of 2 scenes with a total of 23 cameras. The evaluation python files are offered by CityFlow at the MTMCT level using three metrics: IDF1, ID Precision (IDP), and ID Recall (IDR).

### 5.2. Training

The ReID component in Electricity model is initialized with ImageNet-pretrained ResNet-101 weights and fine-tuned on the CityFlow and Syntheicle dataset.

#### 5.2.1. Dataset Preprocessing

Since the CityFlow and Syntheicle raw datasets were used for multi-object tracking tasks, their data structures consist of a video file (vdo.avi) and its frame-by-frame annotation file (gt.txt) corresponding to each camera. This does not directly meet the format requirements of image-level datasets for ReID training. Therefore, we had to preprocess and reconstruct the raw data.

First, based on the frame-by-frame vehicle bounding box information provided in gt.txt, we cropped the vehicle targets in the video, named, and saved the cropped vehicle images according to the standard format "vehicle ID.c camera ID.frame number.jpg", thus constructing the image\_train dataset for ReID training. Second, since the CityFlow dataset does not provide official ReID query and gallery partitions, we followed common practices for ReID tasks

by automatically generating a validation dataset from `image_train`. For each vehicle ID, one image was selected as `image_query`, and all other images were used as `image_test` (gallery) for performance evaluation during training. Simultaneously, during the data construction process, we ensured that the same vehicle ID appeared at least once in multiple different camera viewpoints, thus meeting the basic requirement of cross-camera vehicle re-identification.

### 5.2.2. Training Using Synthehicle

The ReID model trained on the Synthehicle dataset shows a clear and stable improvement during training. We used Town01-N-night in Synthehicle train set as train dataset. Table 1 reveals that the performance increases quickly in the early epochs and becomes stable after about 15 epochs. As our final results, the model achieves 77.2% mAP and 64.1% Rank-1 accuracy, which indicates that it can correctly match most vehicles across different views in the synthetic environment. However, since the training and testing data both come from the same synthetic domain, this result does not fully represent the model’s performance in real-world scenarios.

Table 1. Model trained Example by Synthehicle: Rank-1 and mAP performance at different epochs.

Epoch	Rank-1 (%)	mAP (%)
1	11.7	8.3
2	18.4	20.1
3	22.3	21.9
4	26.2	29.1
5	26.2	32.4
6	35.9	38.6
7	35.9	41.7
8	36.9	39.4
9	41.7	50.4
10	40.8	49.4
11	52.4	64.2
12	53.4	67.6
13	62.1	71.9
14	59.2	74.2
15	66.0	76.7
16	63.1	76.6
17,18	64.1	77.2

### 5.2.3. Training Using CityFlow

In the training portion of this experiment, the ReID module of the ELECTRICITY framework was trained using the CityFlow dataset, which includes 95 vehicle IDs, 34,537 training images, and 5 cameras. A ResNet-101 pre-trained on ImageNet was used as the backbone network, and cross-entropy loss and triplet loss were introduced for optimization during training. Model parameters were automatically

saved at each epoch, and training was stopped at the fourteenth epoch.

From Table 2, the Rank-1 accuracy is only 3.2% in the first epoch, but quickly increases to 72.6% at the third epoch and 95.8% at the fifth epoch. The model converges after the ninth epoch with Rank-1 reaching 100.0% and mAP exceeding 98%. Since the validation set is constructed from the training data, nearly perfect performance indicates overfitting rather than true generalization ability.

Table 2. Model trained Example by CityFlow: Rank-1 and mAP performance at different epochs.

Epoch	Rank-1 (%)	mAP (%)
1	3.2	4.1
2	40.0	27.8
3	72.6	60.4
4	89.5	78.0
5	95.8	83.8
6	96.8	94.6
7	97.9	94.4
8	96.8	96.4
9	100.0	98.3
10	98.9	96.4
11	100.0	99.7
12	100.0	99.9
13–15	100.0	100.0

## 5.3. Evaluation

### 5.3.1. Evaluate Pretrained Model

We first evaluated the pretrained ELECTRICITY model on the Synthehicle dataset without fine-tuning. We chose four datasets from the training set in different time and weather conditions. Since the Synthehicle official evaluation rules of dataset follows the MOTChallenge[2, 9, 13] format, all tracking results produced by the pretrained ELECTRICITY model were first converted to the standard MOT-style format, where each tracking record is represented as  $\langle \text{frame}, \text{id}, \text{bb\_left}, \text{bb\_top}, \text{bb\_width}, \text{bb\_height}, \text{conf}, x, y, z \rangle$ . The converted results are evaluated using the official Synthehicle evaluation python files based on `motmetrics`, and evaluation metrics are IDF1, ID Precision(IDP), ID Recall(IDR), Recall, Multi-Object Tracking Precision(MOTP), and Multi-Object Tracking Accuracy(MOTA).

As shown in Table 3, in the Town01-N-night scenario, when the pre-trained ELECTRICITY model was tested on the Synthehicle dataset, it achieved an IDF1 value of 0.7485, indicating that overall identity consistency remained at a high level in multi-camera scenarios. However, the values of IDP (2.0100) and IDR (1.5141) show anomalies, both exceeding 1, with a MOTA of -1.9537.

Table 3. MTMC Evaluation Results on Night-Time Scenes with Different Model Types

Scene	Model Type	IDF1	IDP	IDR	Recall	MOTA	MOTP
Town01-N-night	Pretrained	0.7485	2.0100	1.5141	0.5502	-1.9537	0.2550
Town03-O-night	Fine-tuned	0.3968	1.3499	0.3533	0.2197	-0.3442	0.2469

Table 4. MTMC Evaluation Results on the CityFlow Validation Set Using the Trained ELECTRICITY Model. Cam6 to Cam9 in S02 means we use validation dataset in Scene02, with camera6 to camera9

Validation Dataset	IDF1 (%)	IDP (%)	IDR (%)
Cam6 to Cam9 in S02	19.25	17.33	21.65
Cam10 to Cam18 in S05	11.51	9.31	15.06
Cam19 to Cam23 in S05	8.21	5.12	20.80

We speculate that the main reason for the anomalies in IDP and IDR is related to the multi-camera temporal stitching strategy used in Synthehicle’s official evaluation protocol. Using this strategy, video sequences from different cameras are sequentially concatenated into a long sequence over time for unified evaluation. When there are many cross-camera identity associations, identity matching statistics are accumulated multiple times globally, causing the counting methods of IDTP, IDFP, and IDFN to deviate from the standard definition for a single camera. Consequently, the values of IDP and IDR may exceed 1. In contrast, IDF1, as a unified indicator that comprehensively considers identity accuracy and recall, is more robust to the above anomalies and is therefore more suitable as the main evaluation criterion for measuring global identity consistency across multiple cameras in this experiment.

### 5.3.2. Evaluate Model Trained by CityFlow

We evaluated the performance of the ELECTRICITY model in epoch 9, trained on CityFlow and analyzed the results using the official MTMCT evaluation tool. Due to the limitations of our GPU device’s performance, we cannot input video from more than 5 cameras at once, as this would crash our device. Therefore, we divided the 19 cameras in S05 of the CityFlow validation into groups of 4 or 5 cameras.

Table 4 show that the model achieved 19.25% IDF1, 17.33% IDP, and 21.65% IDR in the CityFlow validation scenario02, which is the highest evaluation result. These results indicate that after training on the CityFlow data, the model possesses a ability to associate identities across cameras, but there is still significant room for improvement in overall performance.

### 5.3.3. Evaluating The Model Trained by Synthehicle

We evaluated the performance of the ELECTRICITY model in epoch 18 with the Town03Onight scene, using the offi-

cial MTMC evaluation protocol. As shown in table 3, the model achieved an IDF1 of 0.3968, an IDP of 1.3499, and an IDR of 0.3533. The relatively low IDF1 and IDR indicate that the model still has clear limitations in maintaining consistent identities across different cameras. The recall is only 0.2197, which means that a large number of targets are missed during tracking. We think this is mainly caused by the challenging night-time conditions, such as low illumination and severe occlusion. The MOTA value is -0.3442, which further indicates that the overall tracking performance in this night scene is still unsatisfactory.

In terms of localization accuracy, the MOTP reaches 0.2469. This suggests that when the detection boxes are correctly matched, the position estimation of the targets is relatively stable.

In conclusion, the evaluation results demonstrate that the Town03-O-night multi-camera night scene remains very challenging for our fine-tuning model.

## 6. Conclusion

In this project, we fine-tuned and evaluated the ELECTRICITY multi-camera vehicle tracking model using both the CityFlow and Synthehicle datasets. The main goal of this work was to study the MTMCT task and analyze the performance of an existing MTMCT framework under both real-world and synthetic scenarios. We evaluated the model using standard metrics, including IDF1, ID Precision (IDP), and ID Recall (IDR).etc. The experimental results show that the fine-tuned model is able to achieve basic cross-camera vehicle identity association. However, the overall performance is still limited due to challenges such as large appearance variations and complex traffic conditions.

In addition to the technical results, this project is also our first in-depth experience with machine learning and multi-camera tracking tasks. During the project, we encountered many challenges in dataset preparation, model training, and performance evaluation, but we gradually overcame these difficulties through continuous debugging and experimentation. Through this process, we gained a clearer understanding of the MTMCT pipeline and practical experience in applying machine learning techniques to real-world vision problems.

In summary, we successfully reproduced and fine-tuned the ELECTRICITY framework and provided an experimental analysis of its performance on different datasets. In fu-

ture work, we plan to further investigate the causes of the evaluation errors, improve the ReID module, increase the size of our training and evaluation dataset and expand the training dataset from a single situation to multiple weather scenarios, such as day and rain.

## 7. Team Contribution

### 7.1. Yuexin Chen

Work on writing parts of the slides, report, all code implementation and demo during the project:

- Found and prepared the CityFlow dataset and the Synthehicle dataset.
- Selected the ELECTRICITY model as the experimental framework and understood its structure.
- Ran experiments using the pre-trained model on the Synthehicle dataset and evaluated the results.
- Fine-tuned the model using both the CityFlow dataset and the Synthehicle dataset.
- Evaluated the fine-tuned model using standard evaluation metrics.

### 7.2. Samanwita Das

Worked on developing draft of slide deck, wrote the first three sections of the report, generated sample data using CARLA, and demo during the project:

- **Automated Data Generation:** Developed a robust Python pipeline using the CARLA Simulator API, implementing synchronous mode to ensure frame-perfect synchronization between physics and rendering. Integrated domain randomization logic to automatically vary weather conditions and urban layouts, creating a diverse synthetic dataset for model training.
- **Presentation & Demo:** Co-developed the final presentation deck, specifically designing the introduction, background, and motivation sections.
- **Report & Research:** Authored the Introduction, Background, and Related Work sections, conducting a comprehensive literature review to contextualize our project within current state-of-the-art methods. Synthesized academic citations to clearly establish the theoretical motivation for using synthetic data in autonomous driving tasks.

### 7.3. Luke Waind

Worked on setting up the GPU and environment for working on the project, the middle portion of the slide deck, and editing the report.

- Set up the GPU server we utilized for the project and worked with the IT department to save our data when it crashed.
- Supplemental help with environment setup and generating evaluation matrices from ELECTRICITY output.

- Drafting the middle third of the slides shown in the presentation
- Rewriting portions of the report and checking for mistakes.

**Acknowledgement.** This work builds on the ELECTRICITY framework, the Synthehicle and CityFlow datasets, the CARLA simulator, Mask R-CNN, DeepSORT, and the MOTChallenge evaluation metrics. We would like to thank the open-source communities and dataset authors for enabling reproducible research in multi-camera vehicle tracking and synthetic data generation. We also sincerely thank Professor Thi Hoang Ngan Le for her guidance and support throughout the project.

## References

- [1] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pages 3464–3468, 2016. 2
- [2] Patrick Dendorfer, Hamid Rezatofighi, Anton Milan, Javen Shi, Daniel Cremers, Ian Reid, Stefan Roth, Konrad Schindler, and Laura Leal-Taixé. Mot20: A benchmark for multi object tracking in crowded scenes. *arXiv preprint arXiv:2003.09003*, 2020. 5
- [3] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017. 1, 2
- [4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 770–778, 2016. 2
- [5] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017. 2
- [6] Fabian Herzog, Junpeng Chen, Torben Teepe, Johannes Gilg, Stefan Hörmann, and Gerhard Rigoll. Synthehicle: Multi-vehicle multi-camera tracking in virtual cities. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1–11, 2023. 1, 2
- [7] Fabian Herzog, Johannes Gilg, Philipp Wolters, Torben Teepe, and Gerhard Rigoll. Spatial-temporal multi-cuts for online multiple-camera vehicle tracking. *arXiv preprint arXiv:2410.02638*, 2024. 2
- [8] Hongchao Li, Jingong Chen, Aihua Zheng, Yong Wu, and Yonglong Luo. Day-night cross-domain vehicle re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12626–12635, 2024. 2
- [9] Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. Mot16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*, 2016. 5
- [10] Yijun Qian, Lawren Yu, Wenhe Liu, and Schuster He. Electricity: An efficient multi-camera vehicle tracking system for intelligent city. In *Proceedings of the IEEE/CVF Conference*

on Computer Vision and Pattern Recognition (CVPR) Workshops, pages 582–583, 2020. [1](#)

- [11] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 779–788, 2016. [2](#)
- [12] Zheng Tang, Milind Naphade, Ming-Yu Liu, Xiaodong Yang, Stan Birchfield, Shuo Wang, Ratnesh Kumar, David Anastasiu, and Jenq-Neng Hwang. Cityflow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8797–8806, 2019. [1](#), [2](#)
- [13] Paul Voigtlaender, Michael Krause, Aljosa Osep, Jonathon Luiten, Berin Balachandar Gnana Sekar, Andreas Geiger, and Bastian Leibe. Mots: Multi-object tracking and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7942–7951, 2019. [5](#)
- [14] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pages 3645–3649, 2017. [2](#)
- [15] Yue Yao, Liang Zheng, Xiaodong Yang, Milind Naphade, and Tom Gedeon. Simulating content consistent vehicle datasets with attribute descent. In *European Conference on computer vision*, pages 775–791. Springer, 2020. [2](#)