IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 30 November 2022, accepted 8 December 2022, date of publication 12 December 2022,
date of current version 22 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3228735

RESEARCH ARTICLE

Multi-Source Feature Fusion for Object Detection Association in Connected Vehicle Environments

SAMUEL THORNTON, BRYSE FLOWERS, (Student Member, IEEE), AND SUJIT DEY, (Fellow, IEEE)
Department of Electrical and Computer Engineering, University of California at San Diego, San Diego, CA 92092, USA
Corresponding author: Samuel Thornton (sjthornt@ucsd.edu)

This work was supported in part by the Center for Wireless Communications, UC San Diego; and in part by the Smart Transportation Innovation Program (STIP).

ABSTRACT Improvements in vehicular perception systems over the last decade have enabled new levels of safety and awareness in modern production vehicles. However, achievable performance of these perception systems is bounded by sensor limitations, such as range, and environmental factors, such as occlusion. Collaborative perception circumvents these limitations by incorporating sensor data from multiple sources to fill in perception gaps experienced by an individual sources' sensors. This paper explores one important aspect of collaborative perception: simultaneously associating objects detected by multiple individual vehicles with each other. This task is crucial as the inability to perform such object association accurately results in duplicate or missed detections, which can lead to unsafe driving behavior. This work proposes a graph neural network model for this task that achieves an average precision (AP) of 0.882 in a challenging virtual environment consisting of 25 unique, simultaneous, and mobile viewpoints. A simpler real-world scenario with two static viewpoints is also evaluated where the model achieves an AP of 0.998, showing that this model can readily transfer to real-world scenarios as well.

INDEX TERMS Intelligent transportation systems, connected vehicles, data fusion, computer vision, machine learning, neural networks.

I. INTRODUCTION

Intelligent transportation systems (ITS) have been evolving at a rapid pace over the last decade. Many production vehicles come standard with an array of sensors, both inside and outside of the vehicle, that allow the vehicle to perceive what is going on inside the cabin as well as in the area surrounding the vehicle. These sensors enable vehicle safety systems such as the advanced driver-assistance system (ADAS), which can give alerts to the driver about dangers on the road or in some cases even have the vehicle drive itself.

Increased vehicular perception is needed for self-driving vehicles to reach higher levels of automation as well as to improve the safety of vehicles with human drivers. Even the Waymo One [1], which is considered to have level 4 autonomous driving capabilities as defined by the Society of Engineering's (SAE) 6 levels of driving automation [2], still does not have a perfect perception of its surroundings and can run into problems when it is trying to navigate through the world such as in occlusion scenarios in where an object, e.g. a building or other vehicle, is blocking its sensors from seeing oncoming vehicles. There is no perfect sensor array that will allow the vehicle to see everything; even the most sophisticated combination of sensors will have some gaps in perception due to limitations of the sensors or factors beyond the sensors control. However, by having vehicles communicate with each other and share sensor data, these gaps in perception from one vehicle can be filled in by another. This can be enhanced even further by the inclusion of street infrastructure sensors as well.

Vehicular networking is becoming reality through emerging deployments of vehicle-to-everything (V2X) communication systems such as cellular V2X (C-V2X) [3] and dedicated short-range communication (DSRC) [4]. These networks will provide the communication resources needed to enable this vehicular sensor sharing. Yet, sensor data sharing brings

The associate editor coordinating the review of this manuscript and approving it for publication was Kumaradevan Punithakumar.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

about its own challenges in how to efficiently fuse data from multiple sources. This paper solves one aspect of this problem: the ability to simultaneously associate object detections from an arbitrary number of viewpoints with high precision. Object detection association is defined as positive for object detections of the same object and negative for object detections of distinct objects. Determining these associations is crucial as inaccuracies can lead to duplicate detections, causing distrust or noise in the system, or missed detections, leading to misinformed driving behavior. This paper focuses on RGB object detection associations due to the widespread usage of RGB cameras on modern vehicles.

In street environment scenarios, creating positive object associations is difficult as the viewpoint differences for each vehicle's sensors can be quite large. For large viewpoint differences, it is likely that the vehicles will be viewing different surface areas of commonly detected objects which results in a lack of overlapping visual features. However, these positive associations need to be established in order to avoid duplicate detections from appearing for the same object in the fused environmental awareness. An example of this type of association can be seen in Fig. 1, where the truck in the middle of the intersection can be seen by multiple other vehicles. The large viewpoint differences cause the orientation, lighting, and scale to be vastly different amongst each detection even though they are detecting a common object. There are also cases where multiple vehicles view objects that each other cannot see due to being out of one or more vehicle camera's field of view; in this case a negative association would need to be established so that each vehicle can be made aware of the objects they were unable to see individually – enhancing everyone's environmental awareness.

This work shows that object detection association from multiple sources is feasible to do in the real world, in real time, but requires both position estimates and visual descriptors for good performance. This paper proposes a new method to solve this object association problem even in cases where there is no overlap in the visual features. The ability for our proposed model to work for any combination, and any number, of camera viewpoints is one of the key aspects that separates this from related works. More specifically, the contributions of this paper are as follows:

1) We propose a collaborative perception framework that utilizes both multi-source and multi-modal data to create visual and spatial descriptors for achieving real-time sensor fusion in connected vehicle environments.
2) We created and implemented a novel neural network-based machine learning model for accomplishing object detection association that consists of a convolutional feature extractor, graph neural network feature refiner, and a fully connected classifier.
3) We generate a collaborative sensing dataset that contains a large virtual dataset as well as a smaller real-world dataset to validate our model; to address the lack of available collaborative sensing datasets, we are releasing both datasets to the community for future research.
4) We present the results of our proposed model on the generated datasets: an average precision (AP) of .882 on the virtual test set and an AP of .998 on the real-world test set.

The remainder of the paper is organized as follows. In Section II we review related work and the differences between those and this paper. In Section III we give more detail on collaborative perception and object detection association before formulating the problem definition and presenting the framework developed to solve this task. Section IV describes methodology for generating the two datasets utilized in this work. We present results in Section V on both the virtual and real-world test sets, provide a comparison to other methodologies, and provide the computational inference time of our model to establish its ability to execute in real time. Section VI presents the results of a number of ablation studies that informed design choices in the model architecture and demonstrate the final model's robustness to sensor noise. Finally, we conclude the paper in Section VII and share our goals for future work.

II. RELATED WORK

While vehicular perception and object association tasks in general are well studied, the specific task undertaken in the current work, association of objects detected in multiple geographically separate views at the same instant in time, remains understudied - likely due to a lack of publicly available datasets. This task is distinct from related works where public datasets are available: i) Vehicle Re-Identification [5] associates vehicles seen from multiple static views, but at different time instances (limiting the ability to incorporate spatial awareness shown to be highly beneficial in the current work) and typically operates in a surveillance setting (where overlapping visual features nearly always occur due to the high vantage point), ii) Image Retrieval [6] ranks and returns similar images in a corpus from a query image, but again provides no ability to incorporate spatial awareness and does not have a notion of classifying whether two images of are of the same exact object only that they are conceptually related, and iii) Multiple Object Detection and Tracking [7] traditionally considers a single viewpoint and associates current detections with those from prior time steps even if they were momentarily lost due to occlusion. These differences are compared in Table 1 over four attribute categories: mobility

TABLE 1. Comparison between object detection association and other related methodologies.

Attribute	Vehicle Reidentification	Image Retrieval	Multi Object Tracking	Object Detection Association
Mobility				
Multi-Source				
Spatial				
Temporal				

- Always Utilized
- Never Utilized
- Method Dependent

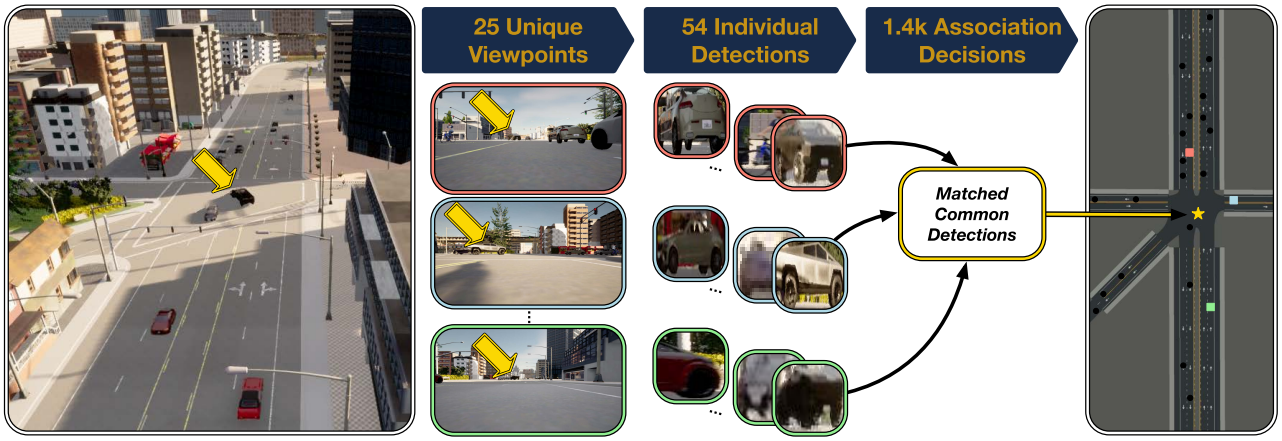

FIGURE 1. Example urban street scenario showing how collaborative perception can improve individual perception. 25 vehicles individually detected 54 objects which leads to 1431 possible detection pairs that can be associated. This figure explicitly shows one common object, a truck in the intersection, being detected by at least three separate vehicles. As can be seen, the vehicles detect the common object at varying scales, lighting conditions, and, most importantly, from nearly orthogonal viewing angles such that there is essentially zero overlap between sections of the detected truck seen by the individual vehicles.

of the data source, use of multiple data sources, use of spatial features, and use of temporal tracklets (data from previous time steps).

The most closely related works have attempted to solve the object detection association problem in a more limited scope. Guo et. al. [8] created a dataset, using the Unity game engine as a simulation environment, where vehicles are randomly placed in an intersection while being observed by two connected vehicles in *fixed* positions at perpendicular sides of the intersection. They create a novel neural network architecture that uses up to four different features in a detection branch to feed pairs of features sets into an association branch which uses a contrastive loss function to predict associations between detections. While they are able to achieve high accuracy, the dataset is limited to two viewpoints which makes the problem much easier than what the current work solves. In our previous work [9], and the real-world evaluation in Section VII, we have also studied this problem from the view of two stationary cameras by exploring two different features sets and machine learning models showing that a neural network model can achieve high accuracy. However, the current work moves beyond this to create a more realistic (and challenging) dataset which simultaneously contains up to 25 different vehicles' perspectives where the potential viewpoints are infinite as the observers are mobile. This work shows that solving this more challenging task requires additional feature sets and higher complexity neural network models.

Liu et. al. [10] proposed a vehicle perception sharing system where two vehicles can have their detections merged into a collaborative view using a bipartite matching algorithm based on the similarity between detection feature sets. However, while real-world data was utilized to validate their method, the dataset is quite limited in scenarios since it only examines the case of two vehicles driving side by side (where viewpoints, and the set of detections for each vehicle, are quite similar); their feature sets include SURF [11] and SIFT [12] descriptors which would not work in scenarios with large viewpoint differences that do no have overlapping visual features. The current work does not require any visual overlap in the common detections of vehicles while still being able to accurately determine object associations. Additionally, the methodology from [10] is inherently only applicable to two viewpoints at a time due to the usage of the bipartite matching algorithm, whereas the methodology presented in this work can associate detections from an arbitrary number of viewpoints.

Prior works have studied cooperative vehicular perception using lidar sensors [13], [14], [15], [16], [17] where the association problem is significantly easier. Furthermore, other works present methodology which does not require machine learning and relies only on the position estimates of surrounding objects to create associations. These works all require lidar and/or radar sensors to estimate object positions [18], [19], [20] whose wide scale adoption in vehicles, especially in the former, remains uncertain due to a variety of reasons while the current work instead focuses on the use of the widely used RGB cameras.

Additionally, a dataset similar to the virtual dataset generated for this work was generated for related research tasks in collaborative perception [21]. While their dataset does include lidar data, it only includes up to 5 vehicles whereas ours contains up to 25 in a single scene.

III. COLLABORATIVE PERCEPTION IN CONNECTED VEHICLE ENVIRONMENTS

As evidenced by the number of works in the previous section, there are many who believe that collaborative perception has great potential for increasing the safety of vehicles, but how this collaborative perception should be implemented has not yet reached a consensus. While sensor fusion will undoubtedly play a large role in any collaborative perception

system, the complexity of sensor fusion that should be used varies widely. For many, this sensor fusion is only looked at for improving individual vehicle perception and involves fusing lidar data with RGB image data to improve the environmental awareness of a particular vehicle [22]; if a point cloud representation of surrounding objects are used then a stitching algorithms can be used to create a panoramic scene with all the provided data. However, these sort of sensor fusion techniques can be computationally expensive and can produce output data that is entirely too large to send over any type of V2X communication channel. For a real-time collaborative perception system, the sensor fusion should be lightweight in order to meet the strict latency requirement for extended sensing in connected vehicle environments. As such object detection association has been chosen as the sensor fusion methodology for this work and will be the topic of the remainder of this section.

A. OBJECT DETECTION ASSOCIATION ON ROADWAYS

While object detection association between vehicles can be utilized to accomplish high level sensor fusion that can be executed in real time, there are a number of challenges in creating a system that can accomplish this task with high accuracy. Vehicles are driving on the roads 24 hours a day, every day of the year, so there are huge variations in the lighting, roadway backgrounds, and weather conditions encountered. Even more challenging is the fact that there are no set viewpoints, so for a given pair of vehicles there is no guarantee that there will be any common visual features even if they are viewing the same object; examples of this can be seen in Fig. 2. As seen in the figure, there are cases of the same vehicle that look quite different due to the different viewpoints and lighting conditions and there are some cases of different vehicles that may look similar to each other. Solutions to this problem must be robust to all of these sources of noise, scale to a larger number of simultaneous view points, and not require overlapping visual features.

B. PROBLEM FORMULATION

The current work restricts itself to object classes of road users such as cars, trucks, buses, motorcycles, and bicycles and purposefully ignores static classes of objects that may appear on the roads such as traffic lights, traffic signs, and benches/chairs. Additionally, the class of pedestrians will not be considered in this work as the feature extraction and classification needed for this object class is different than that of vehicles and we plan to address this problem in future work.

For this object detection association problem, it is assumed that each participating vehicle can provide multi-modal sensor data that is time synchronized. The first type of data is image data; some cars have many external RGB cameras, however, only a front facing RGB camera will be considered for simplicity in this work. While this is principally an image association problem, vehicle telemetry data is used as well, specifically position and orientation estimates for the vehicle that are synchronized with the RGB camera images, due to the benefits this type of data gives to the association performance (which is detailed in Section VI). The last type of data our system utilizes is depth data; the vehicle could sense the depth of surrounding objects using a depth sensor, such as radar or lidar, or if there is no such on board sensor, the depth can be estimated from the sequence of RGB images received (as in done in Section IV-C). The physical transform, from the vehicle position to the sensors positions, must be known so data can be transformed into a common coordinate system. Putting all of this information together creates the input for our proposed framework as shown in Fig. 3. Here we define what we call a *frame*, which is the data being considered for a single time step. The data in one frame is both multi-source, since it is coming from multiple vehicles, as well as multi-modal since it uses multiple types of sensors.

For each time step of data there are k data sources. Each data source detects n_i objects in its surroundings and these detections are concatenated to form the combined set of detections D defined as

$$D = \{d_1, d_2, \ldots, d_N\} \tag{1}$$

$$|D| = \sum_{i=1}^{k} n_i = N \tag{2}$$

along with its cardinality, N, which is the sum of detections from each of the K vehicles. There exists an association between each pair of detections defined as

$$A = \{a_1, a_2, \ldots, a_M\} \tag{3}$$

$$|A| = \binom{N}{2} = M \tag{4}$$

where each detection a_i in A is either positive ($a_i = 1$), meaning that the two detections are of the same object, or negative ($a_i = 0$), meaning that the detections are of distinct objects. The cardinality of A is the number of potential associations, M, and it has a 2-combination relationship with the total number of detections. In this work, we are presenting our model which attempts to predict the correct set of associations. The set of associations, $\hat{A}$, that our model predicts is defined as

$$\hat{A} = f(D) = \{\hat{a_1}, \hat{a_2}, \ldots, \hat{a_M}\} \tag{5}$$

$$|\hat{A}| = M \tag{6}$$

which are obtained by applying our model, $f(\cdot)$, to the set of detections. Since our model is a neural network, $f(\cdot)$ can be thought of as a black box that takes every possible pair of detections within D as input and outputs the predicted association for each pair. Our goal in this work is to maximize the precision of A and $\hat{A}$ as defined by

$$\text{Precision} = \frac{TP}{TP + FP} \tag{7}$$

where TP (True Positives) are cases when a particular $a_i = 1$ and the corresponding $\hat{a_i} = 1$ and FP (False Positives) are cases when a particular $a_i = 0$ and the corresponding $\hat{a_i} = 1$. While precision is the metric of focus in this work, we will

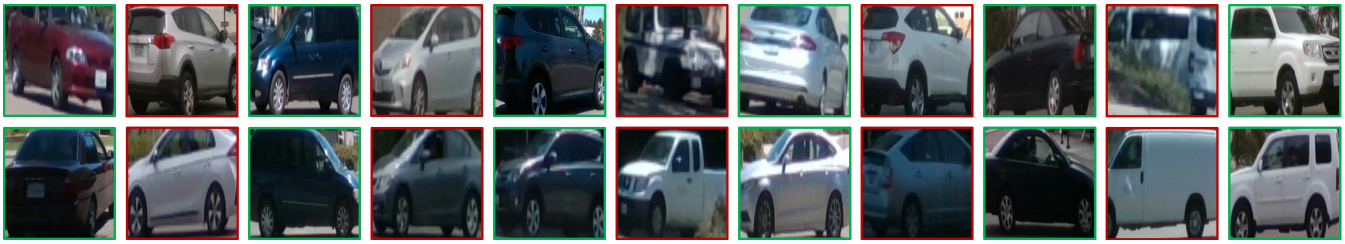

FIGURE 2. Example image pairs for the object detection association problem from the ground truth labels. Green columns represent positive associations, where the two images are of the same underlying vehicle, and red columns represent negative associations, where the two images are of different underlying vehicles.

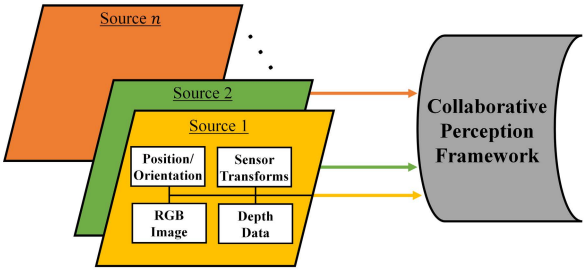

FIGURE 3. Diagram defining what a "frame" of data is in our terminology. Multi-modal data is produced by each data source and the combination of all data sources produce a "frame" of data which is the input to our framework.

be presenting results for a few other metrics as well. One of these is recall which is defined by

$$\text{Recall} = \frac{TP}{TP + FN} \tag{8}$$

where FN (False Negatives) are cases when a particular $a_i = 1$ and the corresponding $\hat{a}_i = 0$. Lastly, there is specificity which is defined by

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{9}$$

where TN (True Negatives) are cases when a particular $a_i = 0$ and the corresponding $\hat{a}_i = 0$.

C. OBJECT DETECTION ASSOCIATION FRAMEWORK

Fig. 4 presents the overview of our proposed framework for determining object detection associations. As mentioned in Section III-A, the input of the framework is a set of multi-modal sensor data from each participating data source, but this data must be pre-processed before being ingested by the fusion model (Fig. 4a). Each RGB image is passed through an object detector to produce a set of region of interest (RoI) images of the objects that each vehicle has detected. Then, each RoI image is transformed into a feature vector. There are two separate types of features that are generated from each RoI image. The first type of features is spatial, represented as a position estimate for the object detected within the RoI image. This position estimate requires all of the input sensor data to generate since the object is detected by the RGB camera, the depth of the object is estimated using the corresponding depth map, and it's position in the world can be estimated using the position/orientation of the vehicle and the physical transform parameters between the vehicle's position/orientation sensors and the RGB camera and depth sensor if the vehicle has one. The second type of feature is visual, which is the result of a feature extractor being applied directly to the RoI image.

The two feature vectors representing each RoI image are concatenated to form the initial features set shown by blue circles in Fig. 4 and serve as the detection set D. Each of these initial features are represented as nodes in a fully connected graph where each edge in the graph represents a potential association between two objects; the set of all edges serve as the set of associations A. At the output of the framework, each edge will need a prediction to determine whether that association is positive or negative but since the graph is fully connected the amount of edges to perform inference on is needlessly excessive. To reduce this, two different types of data filtering are considered in this work (Fig. 4b). The first type is distance filtering, where all edges between objects with estimated positions greater than some threshold distance δ are removed. The second type is source filtering, where edges in the graph between detections that originate from the same source vehicle are eliminated as well under the assumption that the object detector utilized does not produce duplicate detections.

After filtering is applied to the initial graph, a new filtered graph is produced which has significantly fewer edges than the initial graph. This filtered graph will go through one more processing step where each node's features can be refined by selectively aggregating and embedding features from adjacent nodes (Fig. 4c).

The last step of our proposed model is to preform edge prediction for the remaining edges in the graph. A classifier is applied to each pair of nodes where an edge exists and a resultant edge probability is produced (Fig. 4d). This edge probability represents the confidence of our model that the pair of node that edge is between is a positive association, and these probabilities can be thresholded to determine the predicted associations.

D. MACHINE LEARNING MODEL

The visual feature extractor used is a combination of a deep feature extractor and a handcrafted feature extractor; the deep

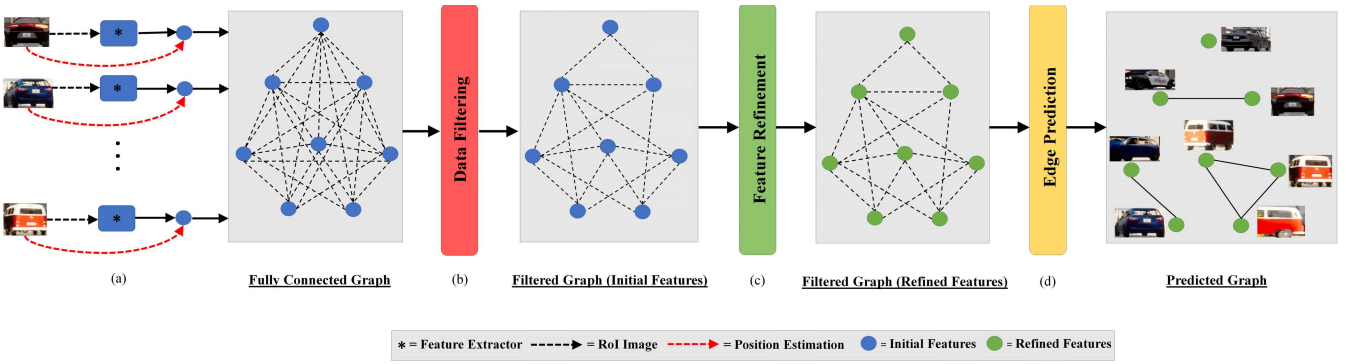

FIGURE 4. Overview of the proposed framework for object detection association. Each node in the graph corresponds to a detected object and contains its visual features as well as position estimates. The graph starts fully connected, but can have many of its edges removed through data filtering. A feature refiner updates the features of each node in the graph, before each pair of nodes that have an edge between them have their features fed into a classifier to predict that edge probability. These probabilities can be thresholded to produce a resultant graph with the predicted links.

feature extractor is ResNet-18 [23] and the handcrafted feature extractor is RGB color histograms.

With our graph representation of this problem, a graph neural network (GNN) becomes a logical option to refine the features of each detection with a more global context as a GNN can simultaneously aggregate across all potential associations at the same time instead of only operating on detection pairs. As such a GNN is used as the feature refiner in our proposed model. The graph neural network that was chosen is the ClusterGNN [24] convolution. The discussion of the ablation study preformed that helped inform these design choices is delayed for further discussion in Section VI. The edge classifier used is a set of three fully connected layers with input size of 1076 and output sizes of 64, 16, and 1 respectively with Exponential Linear Units [25] between the first two layers and a Sigmoid activation function on the output layer; the weights are randomly initialized in each training iteration. The training parameters used are a batch size of 8, a learning rate of .001, the binary cross entropy loss function, and the Adam optimizer [26]. This classifier along with all models implemented in this paper have been implemented in the Python programming language, using the Pytorch and Pytorch geometric deep learning framework as well as the scikit-learn library.

IV. DATASET GENERATION

Since machine leaning is being utilized for this task, the dataset used for training and testing the model is an important decision. As such, we have created our own datasets to represent the scenarios considered in this work. As there is a lack of cooperative sensing data sets for applications in connected vehicle applications, both the synthetic and real-world datasets that we have recorded are available to the public in hopes that they may be useful to other researchers in this field.[1]

[1]The datasets along with a Jupyter Notebook to help get started are available for academic and research purposes; you can request access using the following form: https://forms.gle/EwBrKGWmRywNziVq8

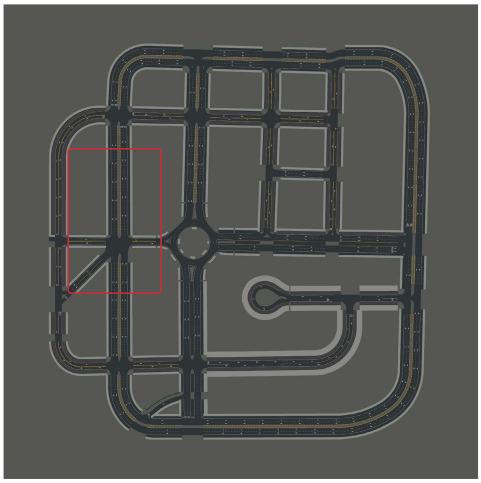

FIGURE 5. Overview of the CARLA simulation environment ("Town03") that was used for synthetic data collection. All data recorded was within the study area shown in the red box.

A. VIRTUAL DATA

While recording a real-world dataset is possible (and is undertaken in section IV-C), it is also time consuming, both for the actual data recording, but especially in the data labeling. To combat this, we created a virtual street environment to generate data that mimics situations encountered in real-world driving scenarios. We used the CARLA [27] autonomous driving simulator to generate an urban street environment where 25 vehicles were randomly placed and instructed to drive around in normal traffic patterns. Since our focus in this work is object detection association, an arbitrary 200 by 92 meter section of "Town03", one such urban street environment provided by CARLA, was selected to act as a study area and the vehicles were restricted to only drive within this area. This was done to maintain a high vehicle density while limiting simulation complexity in order to avoid having many vehicle's cameras record frames that contain no other vehicles or no common vehicles seen by others. An overview of the street environment and study area can be seen in Fig. 5

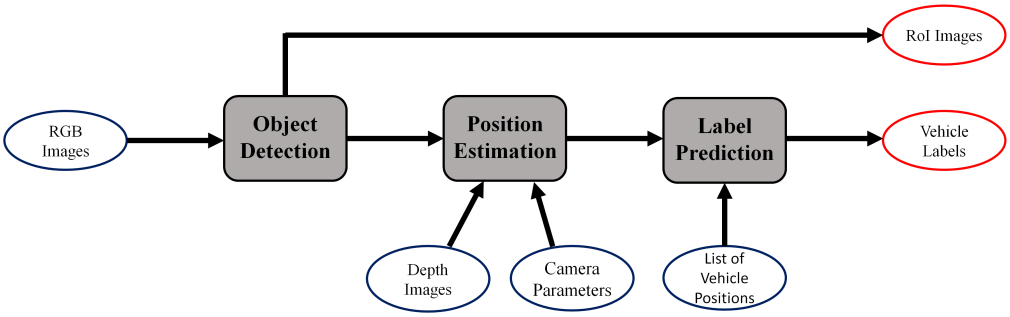

FIGURE 6. Flow graph of the automated labeling pipeline used for the simulation data. It consists of an object detector, position estimator, and label predictor that take ground truth input data from CARLA to produce the output vehicle labels, which all have a corresponding RoI image.

and representative images from the simulation have already been shown in Fig. 1. Each vehicle was equipped with a front facing camera and depth sensor with additional data about the position and orientation of each vehicle also being provided by the driving simulator. 9,080 usable frames[2] were generated in total.

B. AUTOMATED LABELING PIPELINE

9,080 frames corresponds to millions of potential object detection associations that are contained within these frames which all must be labeled as positive or negative associations. Even though CARLA provides ground truth positions and depth images, ground truth is not available for these object detection associations between vehicles. As such, an automated labeling pipeline was created to label this synthetic data. This automated labeling pipeline is an important aspect of the synthetic dataset generation as this is what makes this method of data generation so efficient. There is essentially zero human cost to create this data since it takes no work to generate, just computing time. The pipeline is shown in Fig. 6 and organized as follows: Each image from each vehicle has an object detector applied to it; the object detector used in this work was Detectron2 [28]. As previously mentioned, static detection classes such as traffic lights, stop signs, and benches are filtered out so that only road vehicles like motorcycles, bicycles, cars and trucks remain. After this the camera transform, position of the bounding box in the image frame, and the corresponding depth map (which can be generated by a co-located sensor in CARLA) are used to estimate the 3D position of the detected object in the global coordinate system. Each detection location is then compared with the known ground truth locations of all roadway users in that time step to determine its label. In total, there were 243,365 objects detected, producing 5,185,671 possible association pairs. One thing to note is that this automated labeler utilizes a heuristic and, therefore, is imperfect; for cases of vehicles that are very close to one another it is prone to mislabeling them as the same vehicle. While we have not inspected every vehicle label, 1000 individual labels were randomly sampled and examined to get an estimate of how accurate the automated labeling pipeline is. It was found that the labeling was 93% accurate in this sub-sample, so we have confidence that the large majority of the labels are correct. The utility of this automated labeling procedure is further reinforced by the excellent performance when transferring to the real-world dataset shown in Section VII which is hand labeled, and is thus 100% accurate.

After all the data had been labeled, the 9,080 frames were split into training, validation, and testing datasets that consisted of 7017, 893, 1170 frames respectively; this dataset is named Synthetic (Large). Yet, in order to train models faster during the feature exploration described in Section VI, a random subset of the Synthetic (Large) dataset was utilized to create a smaller training and validation set that consists of 2143 and 99 frames respectively; this dataset is named Synthetic (Small). All models are evaluated on the same test set regardless of what is used during training.

This dataset is, by nature of the problem, very imbalanced due to the high vehicle density in the area leading to many more negative associations than positive associations; only 8.2% of the image pairs in the test set are positive associations and 8.7% in the dataset overall.

C. REAL-WORLD DATA

While there are many benefits in using virtual datasets, one thing they cannot do is perfectly replicate many real-world phenomenon and corner case situations that arise in physical environments. As such, we have also recorded a real-world dataset to test our model that ensures it is able to work in this realistic domain as well. The scenario of this real-world data is RGB camera images from two cameras observing vehicles driving through an intersection as shown in Fig. 7; the cameras were placed as close to the street as possible to try and mimic two vehicles stopped at opposite sides of an intersection. The images from each camera had object detection applied to them [29] and each RoI image pair between the two cameras were labeled by a human to determine if they were a positive association (1) or not (0). This data was recorded for our previous work [9] which did not include spatial information about the vehicles, so some additions to this dataset were

[2]When there are zero common detections amongst multiple vehicles, for instance if they are far apart or face different directions, the frame is discarded.

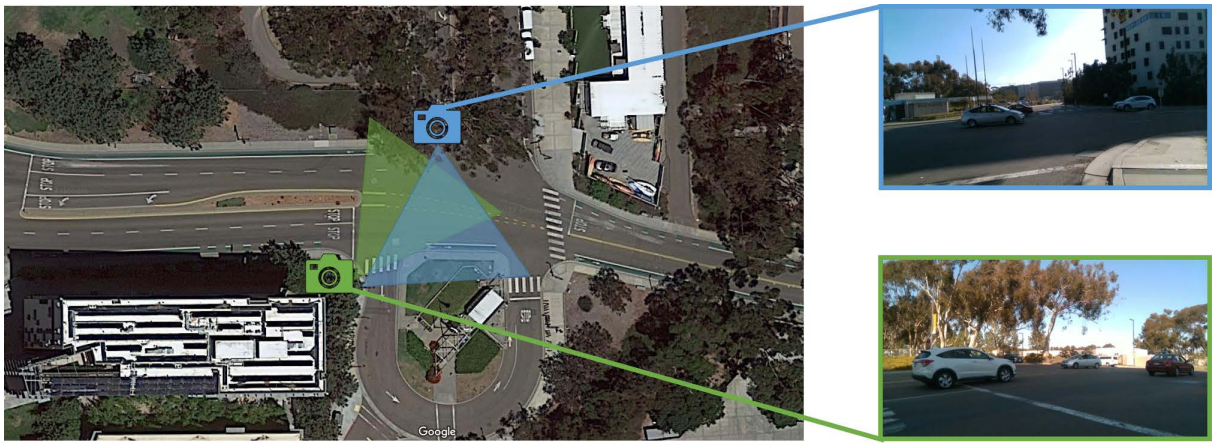

FIGURE 7. The setup used to record the real-world dataset. Cameras were set up on opposite sides of an intersection on the UCSD campus to mimic two cars stopped at the intersection. Example images from both cameras are shown.

needed; these were estimating the transformation between the two cameras as well as creating a depth map of each image, which was done using monocular depth estimation [30]. With these additions, rough position estimates could be generated in the camera coordinate frame and translated to a global coordinate system, thus the data can be tested using the full model developed in the prior section to validate that our formulation can be readily applied in real-world scenarios. 419 frames of data were randomly chosen to use as a test set which yielded 3,354 object detection associations. A training set of real-world data was also created from an additional 1,353 frames of data that yielded another 11,556 data points. This dataset is also imbalanced but not quite as much as the synthetic dataset due to this data being much sparser (less vehicles seen in each image and only two viewpoints); 12.4% of the images pairs contain positive associations.

V. EXPERIMENTAL RESULTS

In this section, the performance results of our proposed model on both the virtual and real-world data sets are presented, as well as sample inference time values to show that it is able to execute in real time. For performance evaluation, the average precision (AP) metric, which is the area under the precision recall curve, is chosen as the metric of focus as this metric relies solely on precision (positive predicted value) and recall (true positive rate). As previously mentioned, the dataset being used is imbalanced due to the nature of the problem and contains an overwhelming majority of true negative examples which will inflate the area under the receiver operating characteristic curve (ROC AUC) metric whereas the AP metric will not be affected by this.

A. PERFORMANCE ON VIRTUAL TESTSET

The results of our model on the virtual test set can be seen in Table 2. In this table there are four rows, each showing the model's performance when a different filtering combination is used. Looking at the distributions of distances between

TABLE 2. Virtual Results.

Filtering Type	ROC AUC	AP	Test Data Reduction (%)	Positive Association (%)
No Filtering	0.990	0.879	0.0	8.2
Distance Only	**0.990**	**0.882**	61.8	21.6
Source Only	0.930	0.834	7.3	8.2
Distance and Source	0.952	0.836	**66.5**	**23.0**

detections for both the positive and negative classes as seen in Fig. 8, it is clear that nearly all of the positive associations have the distance between the detections of less than 20 meters. Therefore for testing the distance filtering threshold δ is set to be 20 meters. To aid in training δ is relaxed to 50 meters as to not reduce the size of the training set by too much. This filtering helps with reducing the search space as well as helps to correct the class imbalance of the data. This can be seen in last two columns of Table 2; any row that contains distance filtering has a high amount of data reduction and an increased positive association percentage over the baseline (no filtering) case. With source filtering, the performance actually decreases on the Synthetic test set and this is mainly due to the automated labeler; there were cases in the automated labeling where two detections from the same vehicle were labeled as the same even though they were of different vehicles causing the source filtering to erroneously reduce the performance on the synthetic data.

Example images that exemplify the performance of our model are shown in Fig. 9. In this figure, examples of all four possible associations results (true positive, false positive, true negative, false negative) randomly chosen from the results on the Synthetic test set are shown. The achieved AP of .882 shows that our model is able to correctly predict associations most of the time producing results shown in Fig. 9b and Fig. 9d but that there are still some cases such as the ones in Fig. 9a and Fig. 9c that the model will get incorrect.

B. PERFORMANCE ON REAL-WORLD TESTSET

As an initial naive test, the model that we trained on the synthetic data was applied directly to the real-world test set;

TABLE 3. Real-World Results.

Filtering Type	ROC AUC	AP	Test Data Reduction (%)	Positive Association (%)
No Filtering	0.976 / 0.999	0.827 / 0.994	0.0	12.4
Distance Only	0.977 / 0.999	0.830 / 0.992	66.8	37.4
Source Only	0.983 / 0.999	0.845 / 0.997	40.2	20.7
Distance and Source	**0.984 / 0.999**	**0.853 / 0.998**	**75.8**	**51.2**

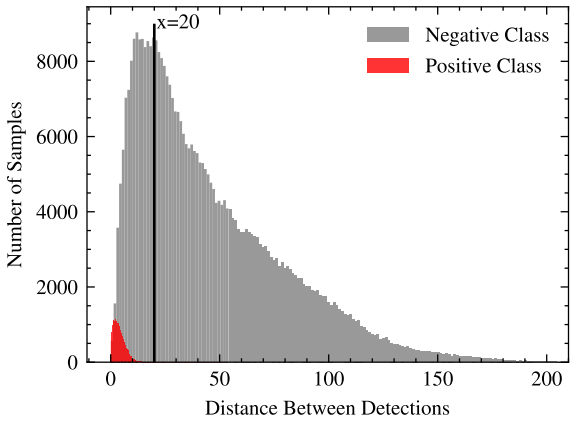

FIGURE 8. Histogram of the distance between detections for both positive and negative association pairs. Almost all positive associations in our Synthetic dataset have an estimated distance between detections of less than 20m.

these are the results on the left side of columns two and three of Table 3. The real-world training set was used for transfer learning as the network model that was trained on synthetic data was fine tuned on this training set for 5 epochs; these results are on the right side of columns two and three of Table 3. Note that there are two different performance values in this section since the results will be different when transfer learning is used, but the percentage of test data reduction and positive associations in the data will remain the same whether transfer learning was used or not.

There is a reduction in AP (0.882 vs 0.830) between the virtual data performance and real-world data performance with the naive test caused by the drastic change between the test data, which is expected since the datasets are from two completely different domains (synthetic vs real world). The transfer learning approach successfully improves the accuracy of the model on the real-world test set and now has a very high accuracy of 0.992 AP when using distance only filtering. Part of why the accuracy on real-world data is higher than the model was able to achieve on the synthetic data is due to the real-world data captured from only two static viewpoints and being more sparse, but nevertheless shows the effectiveness of the model to adapt to different domains.

The combination of source and distance filtering further improves the results on the real-world test set to an AP of 0.998. The source filtering performs better on the real-world data because it was human labeled and thus nearly 100% accurate whereas the synthetic data utilized the automated labeler. However, the real-world performance shows how the use of both distance and source filtering can be an effective method for real-world systems. This combination of filtering produced the greatest reduction in the amount of data that needed to be classified and created the largest increase in the percentage of positive associations in the dataset as seen in the distance and source filtering row of Table 3.

C. MODEL COMPARISON

In terms of model comparisons, a few existing methods were chosen to test on and compare their performance to our proposed model. To the best of our knowledge, there are no existing methods for non-temporal multi-source object detection association using both visual and spatial features but there are other methods that were developed for different applications or use a more limited feature set that can be adapted or applied to this problem. The Siamese-ResNet architecture from our previous work [9] was used to compare the performance of our proposed model with a model that does not use a GNN. While neural networks are the most common machine learning methods for any task that involves digital images, other paradigms can be used; a Random Forest (RF) [31] and a support Vector Machine (SVM) [32] were also used as classifiers to serve as baselines comparison models.[3] For these methods, only the raw object positions are used as features. A method created for this problem that uses a Bhattacharyya Distance Filter (BDF) [19] was also adapted and implemented to determine the associations using a threshold of 3 meters; this method differs from the others in that is does not involve any machine learning. Finally, a different neural network model created for image retrieval [33] (we refer to this model as the Deep Image Retrieval Network (DIRN)) was used as a feature extractor; the cosine similarity between each pair of DIRN features was used as a classification metric. One thing to note is that with the DIRN, no positional features were included which leads to poorer model performance further supporting the results shown in Fig. 11. The results of this comparison are shown in Table 4, showing that our model is the highest performing. Since some methods, like the BDF, produce labels directly (0 or 1) instead of scores (0-1) for each potential association, the binary classification metrics of specificity, precision, recall, and f1 score are the metrics used for comparison instead of AP; classifiers that output scores had their values thresholded to 0.5. The metric of focus here should be the F1 score since this is the harmonic mean of precision and recall. The values for specificity are high for all models but this is mainly due to the dataset containing so many true negative examples, further showing why precision/recall focused metrics work better for this particular problem.

D. COMPUTATIONAL TIME

To demonstrate that our system is feasible to run in real time, different sections of our proposed model were timed to estimate inference times using an NVIDIA 1080Ti GPU. The individual image feature extractor (ResNet-18) took average of 12 milliseconds per image while the GNN layer

[3]The SVM used was sklearn.svm.SVC with kernal=rbf and the RF used was sklearn.ensemble.RandomForestClassifier with n_estimators=100.

(a) False Negatives (b) True Negatives (c) False Positives (d) True Positives

FIGURE 9. Randomly selected image pairs showcasing the results of our model on the Synthetic test set in each of four possible scenarios: (a) false negatives are separate views of the same common object which were not correctly matched, (b) true negatives are separate views of distinct objects which were correctly determined to not be common detections, (c) false positives are separate views of distinct objects which were erroneously matched, and (d) true positives are separate views of common objects which were correctly matched.

TABLE 4. Model performance comparison.

Model	*Specificity*	*Precision*	*Recall*	*F1 Score*
GNN*	**0.989**	**0.831**	**0.765**	**0.796**
Siamese-ResNet [9]	0.985	0.782	0.753	0.768
RF [31]	0.984	0.724	0.596	0.654
SVM [32]	0.975	0.678	0.736	0.706
BDF [19]	0.976	0.678	0.698	0.688
DIRN† [33]	0.984	0.349	0.123	0.181

*Our proposed model
†Does not include object positions

and classifier combined averaged only 2 milliseconds per mini-batch. These inference times should allow for real-time object associations for all but the most dense traffic scenarios. These inference times can be improved by using more powerful hardware or using other techniques, including parallelizing the feature extraction task and the use of road side units and vehicular edge clouds, which look promising to be deployed on roads to support such vehicular applications in the future [34].

VI. ABLATION STUDY

In this section, we will present three separate ablation studies that were conducted as part of this work. The ablation in subsections A and B were done as part of our implementation of our object association framework and helped inform the design decisions for the machine learning model while the ablation in subsection C shows the robustness of the final model to sensor noise.

A. GNN CONVOLUTION

A GNN is a neural network that works directly on a graph structure which consists of nodes and edges; each node contains a feature set and edges represent some sort of relationship between the nodes. There are many different types of graph convolutions present in prior work [35], all of which involve aggregating the features of connected nodes according to some function which varies among methodologies.

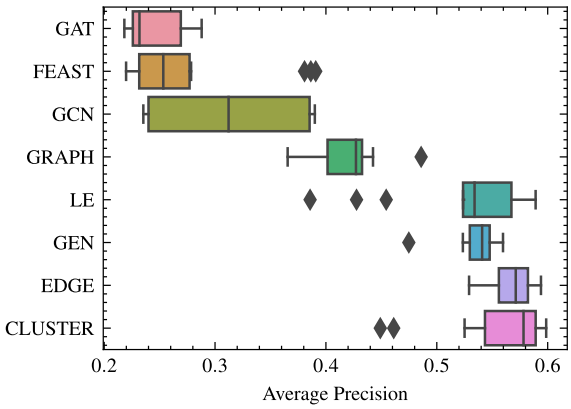

FIGURE 10. Results of the graph convolutional layer accuracy comparison; the results varied greatly depending on the type of layer that was chosen.

After each node in the graph has passed through the GNN layer,[4] we refer to the set of updated node features as GNN features. A number of different graph convolutions were trained on the Synthetic (Small) training set and tested on the Synthetic test set to compare their performance in order to find which would perform best for this problem. While there are many advantages to using a GNN, one disadvantage is that how you aggregate the nodes within the GNN (i.e. what type of GNN convolution you choose) has a large affect on the resultant performance. As such, eight different graph convolutional layers [24], [36], [37], [38], [39], [40], [41], [42] that have been proposed for a variety of other tasks were chosen and uniformly randomly sampled over 100 instances; for each instance, a model with the chosen GNN is trained and the testing results of this experiment can be seen in Fig. 10. ClusterGCN [24] was the best performing graph convolutional layer and as such was the type of

[4]GNNs can have multiple layers, but for this paper all models used that contained a GNN only had 1 GNN layer.

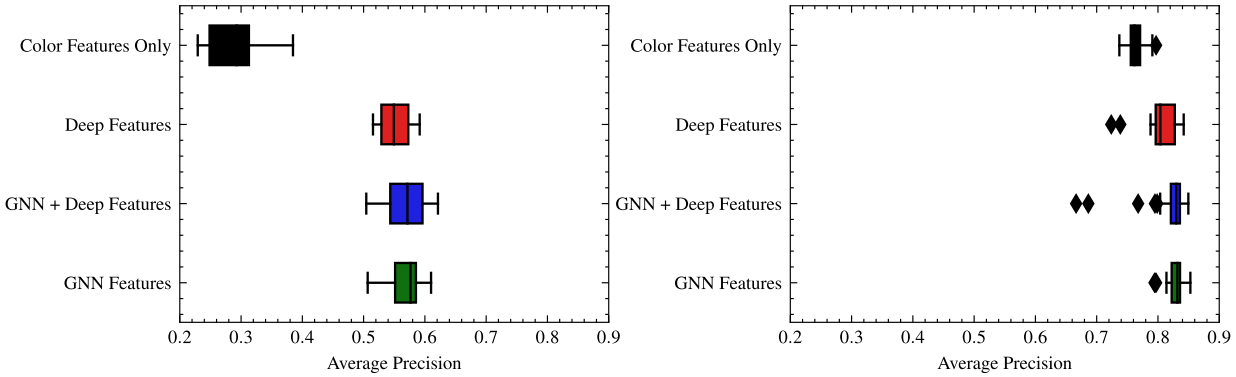

FIGURE 11. Accuracy comparison across multiple classes of potential feature sets and their corresponding models. The results in (a) use only visual features to create the image associations and the results in (b) are when object positions are used as a feature along with visual features. Each combination was repeated numerous times to ensure that variances in training do not impact the takeaways of the analysis.

GNN layer used in our model implementation. Furthermore, as shown in Fig. 11, the GNN models are the highest performing on average, even if only by a small margin.

B. FEATURE TYPE

In this section, we evaluate a few different handcrafted features as well as deep neural network (DNN) and GNN features and compare their performance on this association task.

There are a number of handcrafted features that can be extracted from images that can be used for visual feature association. One of the most widely used methods for this purpose is using a feature extraction method such as SIFT [12] which extracts interest points in images and creates a descriptor for these points that is invariant to scale, orientation, and illumination changes. This type of descriptor acts as a visual feature and it works well for creating associations in images that come from cameras with similar viewpoints and thus have overlapping regions for the descriptors to be associated; however in cases with very large changes in camera pose like in the example of one camera viewing the front of a vehicle and another camera viewing the rear, the interest points detected by SIFT will be of completely different physical points on the vehicles and thus association with these descriptors will fail. As such, more general visual features are needed for this object detection association problem.

One intuitive feature choice is color, since the color of a vehicle is typically uniform. For the specific class of vehicles, you can also try to make observations about what the make, model, and type of the vehicle is. There are machine learning models that exist, such as the Sighthound vehicle recognition API [29], which try to create these make, model, type, and color labels given an image of a vehicle; however, despite the fact that all of these features are intuitive to a human, these models are subject to incorrect classifications. Make, model, and type are specifically very difficult to predict correctly due to the vast diversity of different vehicles on the road today and the amount new vehicles that are released every year. Additionally, even estimating color can be very challenging due to the differences in lighting conditions that are observed in real-world road scenarios. As such, we have decided not to include make, model, and type as specific features for this paper and have chosen more general color features to represent a vehicle color. For each detection, the average of the RGB color channels (3×1 feature vector) is included as well as a color histogram (24×1 feature vector); these *color features* provided to the input of the model to aid in the association classification. Yet, as seen in Fig. 11a, these features when used by themselves lead to subpar performance.

While the handcrafted features presented so far are intuitive and seem like they should work well in object detection association, they have begun to be replaced by deep features in many computer vision applications. Deep features are visual features that are learned by a DNN. These features are usually taken from a feature map of a neural network at the middle or end of the convolutional layers and may not seem meaningful when viewed as images, but they can achieve high accuracy on computer vision tasks. For this paper, we explore the features produced by ResNet-18 [23] pre-trained on the ImageNet [43] dataset. Large residual networks that have been pre-trained on the ImageNet dataset have been shown to work well on a wide variety of image classification problems [44]. As a pre-processing step, all RoI images are resized to $224 \times 224 \times 3$ and have each channel normalized to the ImageNet mean and variance. We take the flattened output of the last convolutional layer of ResNet-18, which has size 512, and call these the *deep features*. As seen in Fig. 11a, incorporating deep features nearly doubles the performance of the model.

Model performance is compared for different combinations of the three visual feature categories (handcrafted features, deep features, GNN features) in Fig. 11a. For these results, a feature exploration experiment was conducted to determine which combination of features perform best on this problem. A list of every combination of features was uniformly sampled to determine what feature set would be used. The feature sets that contain multiple feature categories

are concatenations of each included categories feature vectors. On each iteration a feature set is selected and the corresponding feature extraction model is chosen. Each model is trained on the Synthetic (Small) training set and tested on the Synthetic test set. For each of these feature combinations, the experiment was run for 100 iterations to mitigate the effects of training variance.

So far in this subsection a number of visual features have been discussed, but visual features are only one aspect of this problem; the positional features of each detection can also play an important role in determining if two detections are the same or not. In fact, if absolutely perfect position estimates for each object detection were obtainable, then those alone could be used to create the associations; however, in practice this is not possible. Estimating the position of these objects can be challenging and relies on accurate depth maps that correspond to the images the objects were detected in. There is no real-world depth estimation technique that is perfectly accurate. Even in CARLA, where there is ground truth depth maps available, there is still some error in the position estimation as the depth estimated for a vehicle is for the exterior of the vehicle rather than the centroid; in the case of a large truck or bus this can lead to an error of several meters. As such it is impractical to use position alone to create these associations, especially in dense traffic scenarios, but it is another useful feature to consider in this work. Estimated positions for each detected object can aid in this classification; two objects that have estimated positions that are close to one another are, unsurprisingly, more likely to be the same object while two objects with estimated positions very far away from each other are more likely to be different objects as seen in Fig. 8.

Now that the positions of the detections are being considered, two more features are added to the potential feature set: the raw position of each detection as well as the distance between detections. Considering these additional positional features greatly increases model performance as seen in Fig. 11b

C. SENSOR NOISE

Since the majority of data was generated using a virtual environment with ground truth position/orientation data and depth maps for each vehicle, an ablation study was performed to determine how our model would perform in less ideal conditions which may be more realistic to what may be encountered in the real world. Real-world data was recorded as well and while the results presented in Section VII show good performance on this data, we do not have ground truth for any of that sensor data and as such do not know how much noise in present in each sensor. As such, we have taken the simulation data from the test set and added noise to each measurement needed to estimate a detection's positions (position in meters, depth map in meters, orientation in radians). The noise that was added was sampled from a uniform distribution from $-n_v$ to n_v where n_v is the chosen amount of sensor noise. A range of sensor noises were chosen to reflect the variable nature of these sensors; for many depth sensors, precision is based on the distance to the object so a range of reasonable noise values were chosen [45], [46]. GPS positioning error depends on many factors but is less than 2m in 95% of cases [47] and magnetometers can achieve sub one-degree (0.017 radian) heading (yaw) accuracy [48]. The results of this ablation are shown in Table 5. As is expected, as the sensor noise increases, the AP of the model decreases due to the decreasing accuracy of the position estimates. The model continues to perform well in low to medium noise ranges staying above 0.8 AP; however, with very high noise the model performance will decrease significantly which is to be expected, highlighting the importance of accurate sensor data for this problem.

TABLE 5. Model performance under different levels of sensor noise.

Sensor Noise (position, depth, yaw)	AP
None	0.882
Low (0.3, 0.03, 0.005)	0.879
Medium (1.0, 0.15, 0.05)	0.846
High (2.0, 0.5, 0.2)	0.662

TABLE 6. Model performance under different levels of object detection noise.

Pixel Noise \ Scale Noise	1.0	0.8	0.6
0	0.882	0.867	0.848
10	0.877	0.868	0.851
20	0.860	0.850	0.836

The other source of error that may arise is from poor object detection performance. To simulate this, noise was added to the position and scale of the bounding boxes that were detected by detectron2 [28] to mimic the performance of an object detector on noisier RGB images. Similar to the sensor noise experiment, the noise applied to each value is sampled from a uniform distribution from $-n_v$ to n_v in the case of pixel noise and n_v to 1.0 in the case of scale noise where n_v is the chosen amount of noise. Table 6 shows that even with moderate levels of these type of object detection noise, the model can still maintain an AP of more than 0.8.

VII. CONCLUSION AND FUTURE WORK

In this paper, we proposed the use of object detection association as a lightweight sensor fusion method for real-time collaborative perception in connected vehicles environments. We have presented a novel graph based object detection association framework that leverages both visual and positional features of detected objects. We created a machine learning model for this object association task that was able to produce an AP of .882 on a large virtual dataset consisting of 25 moving vehicle views' and an AP of .998 on a real-world dataset consisting of two stationary views.

The biggest challenges we experienced over the course of working on this paper were figuring out a way to label all the generated virtual data as well as improving the performance of the proposed model on the virtual test set once it was labeled. We were able to develop a heuristic automated labeling pipeline to handle the large amount of data labeling needed and performed ablation studies in feature exploration to determine design improvements in our proposed model.

For future work, we plan to investigate additional real-world considerations for collaborative perception such as high mobility situations, dynamic availability of network and computing resources, and the susceptibility of models to malicious activities such as adversarial attacks. Additionally, while we believe our proposed model performs quite well, there are some further improvements that can be made. For instance, we plan on utilizing 3D object detection instead of 2D to improve the position estimation as well as including temporal tracking to maintain and increase confidence in correct associations while helping to discard incorrect associations. We also plan to consider other classes of objects, such as pedestrians, and introduce other sensor modalities beyond RGB cameras and positional trackers. While this model is able to execute in real time, we believe that it can be further optimized such as amortizing the deep feature extraction (the most computationally expensive task) into actions already being taken by the vehicles (e.g. object detection). Finally, we hope to implement this work in a real time, real-world system using a C-V2X test bed we are developing on the UCSD campus.[5]

REFERENCES

[1] Waymo. *Waymo One*. Accessed: Oct. 6, 2022. [Online]. Available: https://waymo.com/waymo-one/

[2] Society of Automotive Engineers (SAE) International. (Apr. 30, 2021). *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles: J3016_202104*. [Online]. Available: https://www.sae.org/standards/content/j3016_202104/

[3] A. Papathanassiou and A. Khoryaev, "Cellular V2X as the essential enabler of superior global connected transportation services," *IEEE 5G Tech Focus*, vol. 1, no. 2, pp. 1–2, Jun. 2017.

[4] J. B. Kenney, "Dedicated short-range communications (DSRC) standards in the United States," *Proc. IEEE*, vol. 99, no. 7, pp. 1162–1182, Dec. 2011.

[5] S. D. Khan and H. Ullah, "A survey of advances in vision-based vehicle re-identification," *Comput. Vis. Image Understand.*, vol. 182, pp. 50–63, May 2019.

[6] R. Datta, D. Joshi, J. Li, and J. Z. Wang, "Image retrieval: Ideas, influences, and trends of the new age," *ACM Comput. Surv.*, vol. 40, no. 2, pp. 1–60, 2008.

[7] G. Ciaparrone, F. L. Sánchez, S. Tabik, L. Troiano, R. Tagliaferri, and F. Herrera, "Deep learning in video multi-object tracking: A survey," *Neurocomputing*, vol. 381, pp. 61–88, Mar. 2020.

[8] R. Guo, S. Keshavamurthy, and K. Oguchi, "Simultaneous object detection and association in connected vehicle platform," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 840–845.

[9] S. Thornton and S. Dey, "Machine learning techniques for vehicle matching with non-overlapping visual features," in *Proc. IEEE 3rd Connected Automated Vehicles Symp. (CAVS)*, Nov. 2020, pp. 1–6.

[10] H. Liu, P. Ren, S. Jain, M. Murad, M. Gruteser, and F. Bai, "FusionEye: Perception sharing for connected vehicles and its bandwidth-accuracy trade-offs," in *Proc. 16th Annu. IEEE Int. Conf. Sens., Commun., Netw. (SECON)*, Jun. 2019, pp. 1–9.

[11] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," in *Computer Vision—ECCV*, A. Leonardis, H. Bischof, and A. Pinz, Eds. Berlin, Germany: Springer, 2006, pp. 404–417.

[12] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[13] Q. Chen, X. Ma, S. Tang, J. Guo, Q. Yang, and S. Fu, "F-cooper: Feature based cooperative perception for autonomous vehicle edge computing system using 3D point clouds," in *Proc. 4th ACM/IEEE Symp. Edge Comput.* New York, NY, USA: Association for Computing Machinery, 2019, pp. 88–100, doi: 10.1145/3318216.3363300.

[14] Q. Chen, S. Tang, Q. Yang, and S. Fu, "Cooper: Cooperative perception for connected autonomous vehicles based on 3D point clouds," in *Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2019, pp. 514–524.

[15] E. E. Marvasti, A. Raftari, A. E. Marvasti, Y. P. Fallah, R. Guo, and H. Lu, "Cooperative LIDAR object detection via feature sharing in deep networks," in *Proc. IEEE 92nd Veh. Technol. Conf. (VTC-Fall)*, Nov. 2020, pp. 1–7.

[16] T.-H. Wang, S. Manivasagam, M. Liang, B. Yang, W. Zeng, and R. Urtasun, "V2VNet: Vehicle-to-vehicle communication for joint perception and prediction," in *Computer Vision–ECCV*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham, Switzerland: Springer, 2020, pp. 605–621.

[17] Z. Lei, S. Ren, Y. Hu, W. Zhang, and S. Chen, "Latency-aware collaborative perception," in *Computer Vision—ECCV*, S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, Eds. Cham, Switzerland: Springer Nature, 2022, pp. 316–332.

[18] A. Rauch, S. Maier, F. Klanner, and K. Dietmayer, "Inter-vehicle object association for cooperative perception systems," in *Proc. 16th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2013, pp. 893–898.

[19] D. D. Yoon, G. G. Md. Nawaz Ali, and B. Ayalew, "Data association and fusion framework for decentralized multi-vehicle cooperative perception," in *Proc. Int. Design Eng. Tech. Conf. Comput. Inf. Eng. Conf.*, Aug. 2019, Art. no. V003T01A019, doi: 10.1115/DETC2019-98001.

[20] M. Shan, K. Narula, Y. F. Wong, S. Worrall, M. Khan, P. Alexander, and E. Nebot, "Demonstrations of cooperative perception: Safety and robustness in connected and automated vehicle operations," *Sensors*, vol. 21, no. 1, p. 200, Dec. 2020. [Online]. Available: https://www.mdpi.com/1424-8220/21/1/200

[21] Y. Li, D. Ma, Z. An, Z. Wang, Y. Zhong, S. Chen, and C. Feng, "V2X-sim: Multi-agent collaborative perception dataset and benchmark for autonomous driving," *IEEE Robot. Auto. Lett.*, vol. 7, no. 4, pp. 10914–10921, Oct. 2022.

[22] D. J. Yeong, G. Velasco-Hernandez, J. Barry, and J. Walsh, "Sensor and sensor fusion technology in autonomous vehicles: A review," *Sensors*, vol. 21, no. 6, p. 2140, 2021.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, Jun. 2016, pp. 770–778.

[24] W.-L. Chiang, X. Liu, S. Si, Y. Li, S. Bengio, and C.-J. Hsieh, "Cluster-GCN: An efficient algorithm for training deep and large graph convolutional networks," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 257–266, doi: 10.1145/3292500.3330925.

[25] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (ELUs)," 2015, *arXiv:1511.07289*.

[26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[27] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proc. 1st Annu. Conf. Robot Learn.*, 2017, pp. 1–16.

[28] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick. (2019). *Detectron2*. [Online]. Available: https://github.com/facebookresearch/detectron2

[29] A. Dehghan, S. Z. Masood, G. Shu, and E. G. Ortiz, "View independent vehicle make, model and color recognition using convolutional neural network," 2017, *arXiv:1702.01721*.

[30] J. H. Lee, M.-K. Han, D. W. Ko, and I. H. Suh, "From big to small: Multi-scale local planar guidance for monocular depth estimation," 2020, *arXiv:1907.10326*.

[31] T. K. Ho, "Random decision forests," in *Proc. 3rd Int. Conf. Document Anal. Recognit.*, vol. 1, Aug. 1995, pp. 278–282.

[32] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Jul. 1995.

[5] http://cwc.ucsd.edu/research/cellular-vehicle-everything-c-v2x

[33] A. Gordo, J. Almazan, J. Revaud, and D. Larlus, "End-to-end learning of deep visual representations for image retrieval," *Int. J. Compt. Vis.*, vol. 124, no. 2, pp. 237–255, 2017.
[34] L. Liu, C. Chen, Q. Pei, S. Maharjan, and Y. Zhang, "Vehicular edge computing and networking: A survey," *Mobile Netw. Appl.*, vol. 26, no. 3, pp. 1145–1168, 2021, doi: 10.1007/s11036-020-01624-1.
[35] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Jan. 2021.
[36] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017, pp. 1–22.
[37] G. Li, C. Xiong, A. Thabet, and B. Ghanem, "DeeperGCN: All you need to train deeper GCNs," 2020, *arXiv:2006.07739*.
[38] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lió, and Y. Bengio, "Graph attention networks," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–12. [Online]. Available: https://openreview.net/forum?id=rJXMpikCZ
[39] N. Verma, E. Boyer, and J. Verbeek, "FeaStNet: Feature-steered graph convolutions for 3D shape analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2598–2606.
[40] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 1–12, Oct. 2019, doi: 10.1145/3326362.
[41] E. Ranjan, S. Sanyal, and P. Talukdar, "ASAP: Adaptive structure aware pooling for learning hierarchical graph representations," in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2020, vol. 34, no. 4, pp. 5470–5477. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/5997
[42] C. Morris, M. Ritzert, M. Fey, W. L. Hamilton, J. E. Lenssen, G. Rattan, and M. Grohe, "Weisfeiler and Leman go neural: Higher-order graph neural networks," in *Proc. 33rd AAAI Conf. Artif. Intell.*, Jun. 2019, vol. 33, no. 1, pp. 4602–4609. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/4384
[43] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, Jun. 2009, pp. 248–255.
[44] M. Huh, P. Agrawal, and A. A. Efros, "What makes ImageNet good for transfer learning?" 2016, *arXiv:1608.08614*.
[45] K. Khoshelham and S. O. Elberink, "Accuracy and resolution of Kinect depth data for indoor mapping applications," *Sensors*, vol. 12, no. 2, pp. 1437–1454, Feb. 2012. [Online]. Available: https://www.mdpi.com/1424-8220/12/2/1437
[46] Intel. *Intel RealSense Depth Camera D455*. Accessed: Oct. 6, 2022. [Online]. Available: https://www.intelrealsense.com/depth-camera-d455
[47] National Coordination Office for Space-Based Positioning, Navigation, and Timing. *GPS Accuracy*. Accessed: Oct. 6, 2022. [Online]. Available: https://www.gps.gov/systems/gps/performance/accuracy
[48] Avionics Anonymous. *Minimag Magnetometer*. Accessed: Oct. 6, 2022. [Online]. Available: https://docs.avionicsanonymous.com/devices/minimag

SAMUEL THORNTON received the B.S. degree in electrical engineering from the University of Southern California, in 2016, and the M.S. degree in electrical engineering with a focus in intelligent robotics, systems, and control from the University of California, San Diego, in 2020, where he is currently pursuing the Ph.D. degree in electrical engineering.

He was an Undergraduate Research Assistant with the Magnetic Resonance Engineering Laboratory, University of Southern California, from 2014 to 2015. He has been a Graduate Student Researcher at the Mobile Systems Design Laboratory, University of California, San Diego, since 2017. His research interests include machine learning and computer vision with a focus in collaborative vehicular applications.

BRYSE FLOWERS (Student Member, IEEE) received the B.S. and M.S. degrees in computer engineering from Virginia Tech, Blacksburg, VA, USA, in 2014 and 2019, respectively. He is currently pursuing the Ph.D. degree in computer engineering with the University of California, San Diego, CA, USA.

In 2013 and 2014, he interned as an Engineer at Qualcomm, San Diego, CA, USA, later returning, from 2015 to 2017, as an Engineer at Qualcomm, where his work focused on multi-networking for applications, such as VoWifi, hardware accelerated protocol processing, and intelligent network selection. In 2018 and 2020, he interned with the MIT Lincoln Laboratory and HRL Laboratories, respectively. His research interests include the intersection of digital signal processing and machine learning.

Mr. Flowers was awarded the Bradley Masters Fellowship by the Bradley Department of Electrical and Computer Engineering, Virginia Tech, in 2017. In 2018, he received the Hume Graduate Recruiting Fellowship by the Hume Center for National Security and Technology and the Association of Old Crows (AOC) Electronic Warfare Scholarship by the AOC Capitol Club. In 2019, he was also awarded the Powell Fellowship by the Jacobs School of Engineering, UCSD. He was named a Collins Aerospace (formerly UTC Aerospace Systems) Scholar.

SUJIT DEY (Fellow, IEEE) received the Ph.D. degree in computer science from Duke University, in 1991.

In 2004, he founded Ortiva Wireless, where he has served as its founding CEO and later as the CTO and Chief Technologist till its acquisition by Allot Communications, in 2012. Prior to Ortiva, he served as the Chair of the Advisory Board of Zyray Wireless till its acquisition by Broadcom, in 2004, and an advisor to multiple companies, including ST Microelectronics and NEC. He has served as the Faculty Director of the von Liebig Entrepreneurism Center, from 2013 to 2015, and the Chief Scientist, Mobile Networks, at Allot Communications, from 2012 to 2013. In 2015, he co-founded igrenEnergi Inc., providing intelligent battery technology and solutions for EV mobility services. He heads the Mobile Systems Design Laboratory, developing innovative and sustainable edge computing, networking and communications, multi-modal sensor fusion, and deep learning algorithms and architectures to enable predictive personalized health, immersive multimedia, and smart transportation applications. He has created inter-disciplinary programs involving multiple UCSD schools as well as community, city, and industry partners; notably the Connected Health Program, in 2016, and the Smart Transportation Innovation Program, in 2018. Prior to joining UCSD in 1997, he was a Senior Research Staff Member at NEC C&C Research Laboratories, Princeton, NJ, USA. In 2017, he was appointed as an Adjunct Professor at the Rady School of Management and the Jacobs Family Endowed Chair in Engineering Management Leadership. He is currently a Professor with the Department of Electrical and Computer Engineering and the Director of the Center for Wireless Communications and the Institute for the Global Entrepreneur, University of California, San Diego. He has coauthored more than 250 publications, and a book on *Low-Power Design*. He holds 18 U.S. and two international patents, resulting in multiple technology licensing and commercialization.

Dr. Dey has been a recipient of nine IEEE/ACM Best Paper Awards, and has chaired multiple IEEE conferences and workshops.