sensors

MDPI

*Article*

# Decision-Based Fusion for Vehicle Matching

**Sally Ghanem [1,*], Ryan A. Kerekes [1] and Ryan Tokola [2]**

[1] Oak Ridge National Laboratory, Oak Ridge, TN 37830, USA; kerekesra@ornl.gov
[2] Aware, Inc., Bedford, MA 01730, USA; ryantokola@gmail.com
* Correspondence: ghanemss@ornl.gov

**Abstract:** In this work, a framework is proposed for decision fusion utilizing features extracted from vehicle images and their detected wheels. Siamese networks are exploited to extract key signatures from pairs of vehicle images. Our approach then examines the extent of reliance between signatures generated from vehicle images to robustly integrate different similarity scores and provide a more informed decision for vehicle matching. To that end, a dataset was collected that contains hundreds of thousands of side-view vehicle images under different illumination conditions and elevation angles. Experiments show that our approach could achieve better matching accuracy by taking into account the decisions made by a whole-vehicle or wheels-only matching network.

**Keywords:** decision fusion; deep networks; vehicle matching



**Citation:** Ghanem, S.; Kerekes, R.A.; Tokola, R. Decision-Based Fusion for Vehicle Matching. *Sensors* **2022**, *22*, 2803. https://doi.org/10.3390/s22072803

Academic Editors: Jesús García-Herrero and Antonio Berlanga

Received: 25 February 2022
Accepted: 26 March 2022
Published: 6 April 2022



## 1. Introduction

Data fusion is a very challenging topic in machine learning that can provide complimentary information and enhance the understanding of data structures. Multi-modal data have become more widely available due to advances in sensor development. While data fusion is not a new topic, it has recently witnessed a considerable increase in the demand for foundational principles for different applications. There are three levels of data fusion that can be considered, namely, data fusion, feature fusion, and decision fusion. Data fusion involves integrating unprocessed data generated by each sensor. In feature fusion, features extracted from raw data, collected from various sensors, and combined to construct a better feature. Decision fusion, on the other hand, optimally combines the decisions reached by different algorithms or sensors to yield a more informed decision.

The objective of this work was to develop a principled decision fusion framework for vehicle matching using convolutional neural networks (CNNs). The topic of vehicle re-identification has been studied extensively in the literature. In [1], the authors proposed a local feature-aware model for vehicle re-identification. Given multi-view images of a target vehicle, their model focuses on learning informative parts that are most likely to differ among vehicles. However, their model does not perform well on images with a dim backgrounds. Additionally, the model can not achieve effective identification in the case of only different views of two cars or in the absence of shared parts in the two cars. In [2,3], the authors adopted a spatio-temporal approach for vehicle re-identification. Bing et al. [4] proposed a part-regularized discriminative feature preserving method that enhances the ability to perceive subtle discrepancies. Their model utilized three vehicle parts for detection: lights, including front light and back light; window, including front window and back window; and vehicle brand. Oliveira et al. [5] presented a two-stream Siamese neural network that used both the vehicle shape and license plate information for vehicle re-identification. In [6], the authors proposed an end-to-end RNN-based hierarchical attention (RNN-HA) classification model for vehicle re-identification. Their RNN-based module models the coarse-to-fine category hierarchical dependency to effectively capture subtle visual appearance cues, such as customized paint and windshield stickers.

In this work, key features from vehicle side-view images and their corresponding wheels are extracted using Siamese networks. Pattern information specific to the wheels can often provide supplementary information about the vehicles in question; for example, two otherwise identical vehicles of the same make, model, and color, could potentially be distinguished if their wheels or hubcaps are different. Our dataset was collected under various illumination conditions and elevation angles. Individual similarity scores are combined, which are derived from features extracted from either the whole-vehicle images or from cropped images of their wheels. A principled integration of the individual scores is expected to enhance overall matching accuracy; thus, an overall similarity score is reached by a joint aggregation of the whole-vehicle and wheel similarity scores.

The balance of the paper is organized as follows. In Section 2, related work is described. In Section 3, the dataset structure is thoroughly explained. In Section 4, the attributes of our approach are provided and the network structure is defined. In Section 5, our validation along with other experimental results are presented, and Section 6 provides concluding remarks and describes future work.

## 2. Related Work

The topic of multi-modal data fusion has been extensively studied in computer vision. Laying out the fundamentals for data fusion has become crucial for many applications, including target recognition [7–9], handwriting analysis [10], and image fusion [11]. A comprehensive survey of data fusion is provided in [12,13].

Decision fusion techniques can be classified on the basis of the fusion type. The most popular fusion type is voting-based, which includes majority voting, weighted voting, and Borda count, which sums the reverse ranks to perform decision fusion [14]. Other voting techniques are probability-based, such as Bayesian Inference [15] and Dempster–Shafer fusion [16,17]. A detailed comparison of Bayesian inference and Dempster–Shafer fusion is included in [18]. A limitation common to probability based methods is they typically require prior information about sensors' decisions or demand high computational complexity. As a result, their adoption in decision fusion in real-time applications has been negatively impacted. In this work, a decision fusion approach is established that leverages a neural network structure to mitigate the need for prior knowledge or assumptions about the classifiers. The performance of our proposed approach is then compared to that of the better-known majority vote fusion method.

Developing real-time transportation technologies remains a crucial topic for safety, surveillance, security, and robotics. Decisions regarding traffic flow and surveillance need to be performed in real time to detect potential threats and act accordingly. Computer vision systems can be employed to automatically match and track vehicles of interest. In [19], the authors developed a robust framework for matching vehicles with highly varying poses and illumination conditions. Moreover, in [20], the authors proposed a low complexity method for vehicle matching that is robust to appearance changes and inaccuracies in vehicle detection. They represented vehicle appearances using signature vectors and compared them using a combination of 1D correlations. A vehicle matching algorithm was proposed in [21] that identified the same vehicle in different camera sites using color information. Their matching approach took advantage of road color variation to model changes in illumination to compensate for color variation and minimize the false positive matches.

The primary contribution of this paper is a decision fusion framework for vehicle matching which aggregates decisions from two Siamese networks handling a pair of vehicle images and their detected wheels. Integrating the decisions helps reinforce the consistency between the outputs of the matching networks. In our evaluation, a recently collected dataset was used called Profile Images and Annotations for Vehicle Re-identification Algorithms (PRIMAVERA) [22], which has been made publicly available. These data were partitioned into training and validation subsets. After training the vehicle and the wheel matching networks, the learned networks were then utilized to match new observed data

and investigate the generalization power of our approach. Experimental results confirmed a significant improvement in the vehicle matching accuracy under decision fusion.

## 3. Dataset Description

### *3.1. Data Collection*

To substantiate the validation of our proposed approach, a dataset that contains hundreds of thousands of side-view vehicle images was collected. As mentioned above, this dataset has been made publicly available [22]. The data were collected using a roadside sensor system containing one or more color cameras and a radar unit. Three types of cameras were used for vehicle image collections. For daytime image capture, RGB cameras equipped with either Sony IMX290 (1945 $\times$ 1097 pixels) or IMX036 (2080 $\times$ 1552 pixels) sensors were used. For low-light image capture after sunset, a Sony UMC-S3C camera was used to perform high-sensitivity RGB imaging. Images were captured from distances to the vehicle ranging between 1 and 20 m using 1.8 to 6 mm lenses in conjunction with the above cameras. An undistortion operation was applied to each frame prior to any processing in order to remove distortion effects of the wide-angle lenses. Images of passing vehicles were collected over the course of several years. The images that we used in this study were captured during both day and night. The nighttime imagery was captured using a low-light color camera. The sensors were positioned both at ground level and at an elevated angle of approximately 20 degrees from horizontal and were oriented perpendicular to the road, providing a near-profile view of passing vehicles.

License plate readers were collocated with the sensors to provide a ground-truth identity for each collected vehicle. While license plates were used for ground truth, this approach to vehicle re-identification does not rely on a license plate but only a profile view of the vehicle. Actual license plate numbers were obfuscated by replacing each plate number with an arbitrary number that was subsequently used as the vehicle ID. Sample images from the dataset are depicted in Figure 1.

### *3.2. Data Quantity*

The dataset contains 636,246 images representing 13,963 vehicles. Each vehicle has a different number of images depending on the number of collected video frames and how many times it passed by one of the sensors. Our dataset was divided into training and validation sets. The training set contains 543,926 images representing 11,918 vehicles, and the validation set contains 92,320 images representing 2045 vehicles.

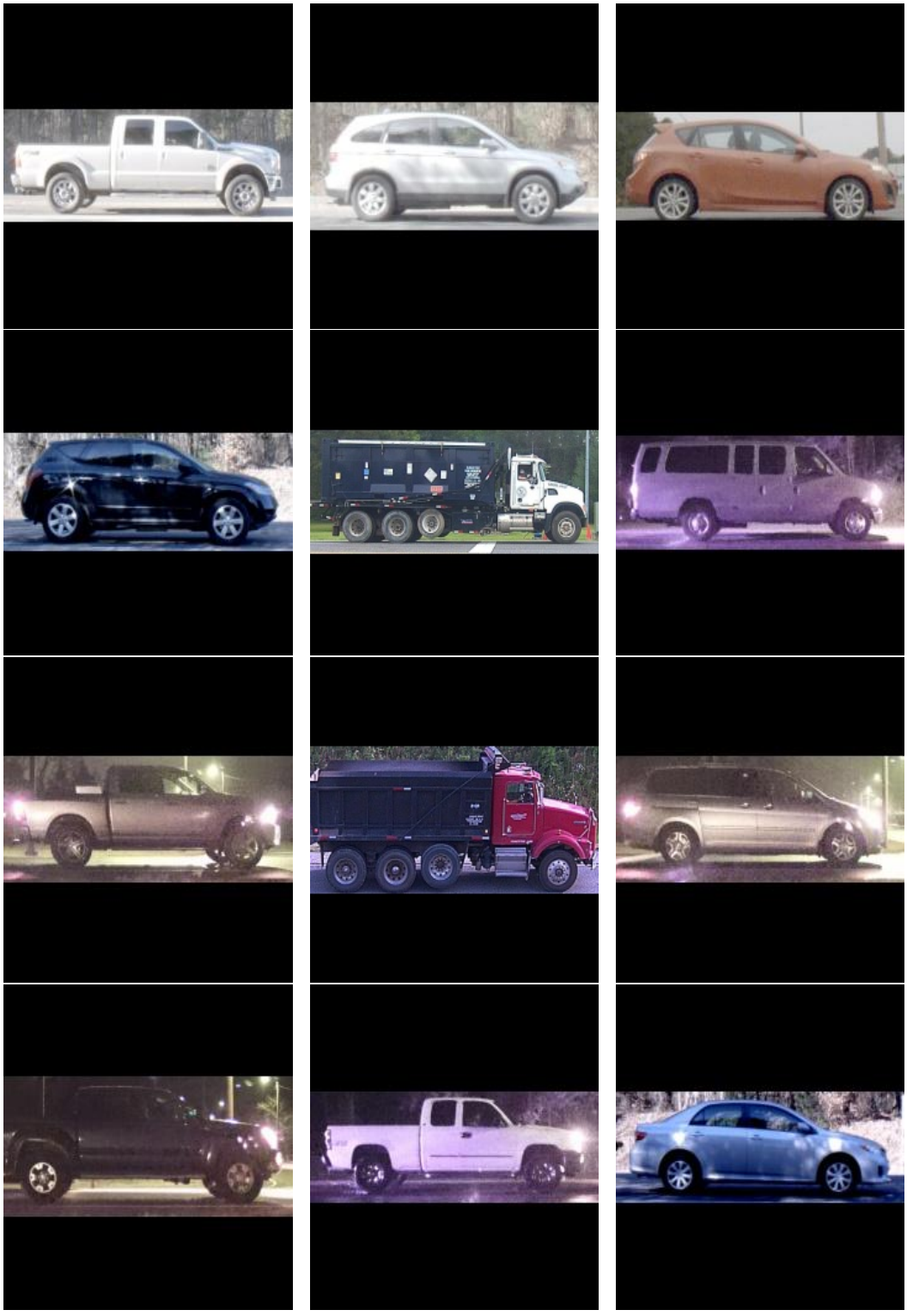

**Figure 1.** Sample images from the PRIMAVERA dataset.

## 4. Network Structure

Our vehicle matching algorithm consists of two main stages. The first stage includes a whole-vehicle matching network, which compares a pair of vehicle images and generates a similarity score. The first stage also contains a wheel detector, which detects the outermost two wheels in each vehicle image and feeds them into the wheel matching network. This network compares the detected wheels and outputs wheel similarity scores. The second stage of the proposed framework is the decision fusion network, which combines the

vehicle and wheel matching scores toward the goal of obtaining a more accurate and robust overall similarity score. The diagram of the proposed framework is depicted in Figure 2.

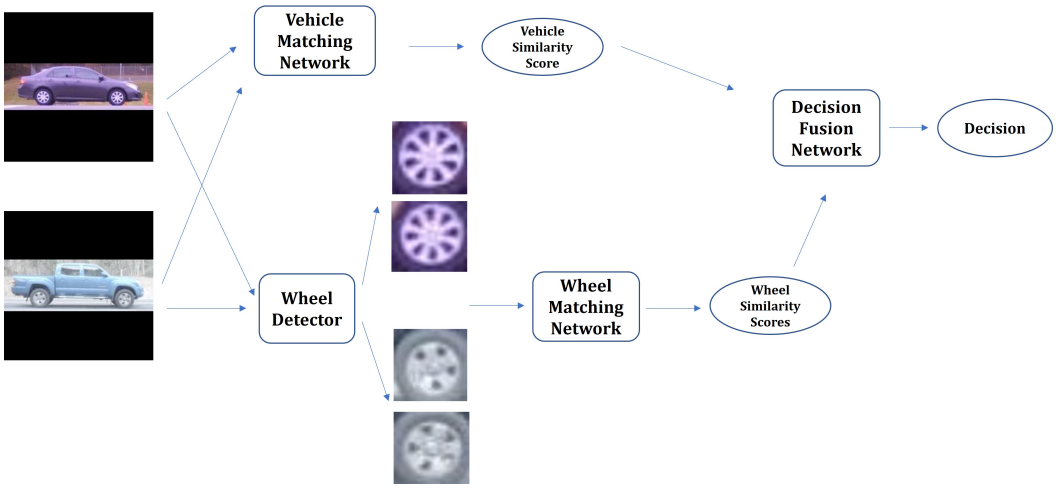


**Figure 2.** Diagram of overall vehicle matching algorithm.

### *4.1. Problem Formulation*

In the following, our problem formulation is explained. Assume having two data points, $\mathbf{X_i^t}$ and $\mathbf{X_{ii}^t} \in R^{m \times n}$ that belong to class $i$ and $ii$, respectively. In addition, consider having $T$ different modalities such that $t$ denotes the modality index and $t = 1, \ldots, T$. More precisely, consider two feature maps $\mathbf{x_i^t}$ and $\mathbf{x_{ii}^t} \in R^c$ corresponding to $\mathbf{X_i^t}$ and $\mathbf{X_{ii}^t}$, respectively. For each modality $t$, a decision $D^t$ is generated such that $D^t = f(\mathbf{x_i^t}, \mathbf{x_{ii}^t})$, where $f$ is the matching function and $D^t \in R$. Our goal is to fuse the decisions $D_{t=1}^T$ so a fusion function $k : D(1), D(2) \ldots, D(T) \rightarrow z$ is defined that fuses the decisions $D_{t=1}^T$ and produces a final decision $z \in R$. Various fusion functions can be used to combine the matching decisions. In this work, the use of neural networks was investigated to better explore the potential contribution of decision fusion. A multi-layered, fully connected decision fusion network was trained that combines the matching decisions and provides a unified, final ruling. In order to show the advantage of using a decision fusion deep learning method, the performance of the decision fusion network is compared to those of other common fusion methods, such as soft voting, majority voting, and averaging the decisions. For soft voting, every individual decision is assigned a probability value. The predictions are weighted by their importance and summed up. The weight that provides the highest training accuracy wins the vote.

The following section elaborates on how the neural networks are constructed for the proposed framework.

### *4.2. Vehicle Matching*

The objective of the whole-vehicle matching network is to provide a similarity score between any pair of vehicle images. For this purpose, a Siamese network [23] was trained [24] using the labeled training set. Siamese network are a class of neural networks composed of two or more identical sub-networks. These sub-networks share the same parameters and weights; in addition, the weights are updated identically during training. The goal of Siamese networks is to find the similarity between the inputs through comparing their feature signatures. One advantage of a Siamese model is that the model does not have to be updated or retrained if an object class is added to or removed from the dataset. Moreover, Siamese networks are more robust to class imbalance, as a few images per class can be sufficient for training.

### *4.3. Wheel Detection*

To explore the contribution of leveraging finer features from the vehicle wheels to the performance of vehicle matching, a wheel detector was trained using a portion of the training data. In particular, the wheels were manually labeled and bounded for 4077 images using LabelImg program [25]. Afterwards, the Single-Shot multibox Detection (SSD) Mobilenet v2 network [26] was retrained to detect wheels using the labeled set. The SSD Mobilenet v2 model is a single-shot detection network developed for performing object detection. The model was trained on the Common Objects in Context (COCO) image dataset [27]. The COCO dataset is geared toward large-scale object detection and segmentation. As the detection stage of the algorithm does not require special considerations, pre-trained CNNs were able to provide sufficient performance with regard to finding a bounding box around the vehicles in each video frame. The subsequently detected wheel positions were used to refine the position of the vehicle within the final cropped image; therefore, high precision is not needed in the initial bounding box coordinates. SSD MobileNet v2 was found to provide adequate precision and recall rates in detecting vehicles for this purpose; as a result, this pre-trained network was used for the detection stage for all experiments in this paper. The Mobilenet V2 model was retrained using the labeled wheel dataset. The dataset was constructed such that the wheel examples were taken from a diverse set of vehicles (e.g., sedans, SUVs, trucks, vans, and big-rigs) under different lighting conditions (e.g., day, dusk, and night). Furthermore, the wheel detector was evaluated on a part of the unlabeled testing set. In Figure 3, some examples of the detected wheels from two different vehicles in the testing set are shown.

**Figure 3.** Two vehicle images and their detected wheels.

### *4.4. Wheel Matching*

As discussed earlier, the wheels can be utilized as an additional source of information; specifically, the aim is to successfully integrate the finer details from the wheel patterns into the overall vehicle matching task and thus achieve better performance in comparison to only using the whole-vehicle images by themselves. In order to match the wheels of each pair of vehicles, another Siamese network was trained to generate wheel similarity scores. The wheel detector that was developed was utilized to estimate the bounding boxes and crop the wheels from the vehicle images. The detected pairs of wheels were saved and later used to train this Siamese network. Our framework is evaluated and experimental results are shown in Section 5.

## 5. Experimental Analysis

In this section, the efficacy of our proposed fusion method is investigated, and its performance is evaluated on the dataset.

### *5.1. Vehicle Matching Results*

As explained in Section 4.2, a Siamese network was trained to match any pair of images and generate a matching score. In our experiment, randomly generated image pairs were taken from a set of 543,926 images, which represent 11,918 vehicles, for training the network. True positive pairs were selected such that the two images were from two different passes of the vehicle (as opposed to two different image frames recorded during the same pass). The vehicle matching network consists of three main components. The first component has two identical branches, and each branch has seven layers. The input to each branch is a single image that has been resized to 234 × 234 × 3. The structure of each branch is depicted in Table 1. A max-pooling step is applied after each layer.

**Table 1.** Vehicle matching network—each branch's structure.

| Layer | Structure | Activation |
|---|---|---|
| Layer 1 | 4 × 3 × 3 | Relu |
| Layer 2 | 8 × 3 × 3 | Relu |
| Layer 3 | 16 × 3 × 3 | Relu |
| Layer 4 | 32 × 3 × 3 | Relu |
| Layer 5 | 32 × 3 × 3 | Relu |
| Layer 6 | 32 × 3 × 3 | Relu |
| Layer 7 | 32 × 3 × 3 | Relu |

The second part of the network is the differencing phase, in which the output of one branch is subtracted from the other. The third and last component of the network is the matching network. This part of the network consists of six layers, and it operates on the differenced input to output a similarity score between 0 and 1. The structure of the matching network is listed in Table 2. These networks were implemented in Python using Tensorflow. Adaptive momentum-based gradient descent method (ADAM) technique [28] was used to minimize the loss functions and apply a learning rate of 0.005.

**Table 2.** Vehicle matching network—matching network structure.

| Layer | Structure | Activation |
|---|---|---|
| Layer 1 | 64 × 3 × 3 | Relu |
| Layer 2 | 64 × 3 × 3 | Relu |
| Layer 3 | 64 × 2 × 2 | Relu |
| Layer 4 | 64 × 1 × 1 | Relu |
| Layer 5 | 32 × 1 × 1 | Relu |
| Layer 6 | 1 × 1 × 1 | Relu |

In our experiments, a binary cross-entropy loss function was utilized for training the network. A batch size of 256 was used for the training stage of the experiments. The network was trained for 100,000 steps, and the performance on the validation set was computed every 100 steps. In order to assess algorithm performance over the widest variety of conditions possible, a diverse subset of the validation set was constructed and used for validation. This subset contains more than 12,000 pairs of vehicle images that were collected at different locations at different times of the day and at different angles of elevation. Since the similarity score has a range of 0 to 1, a threshold of 0.5 was applied to evaluate matching accuracy. If the matching score is more than 0.5, it indicates that the two images belong to the same vehicle, and vice versa. The training and validation

performance of the vehicle matching network is depicted in Figure 4. The results are also shown in Table 3.

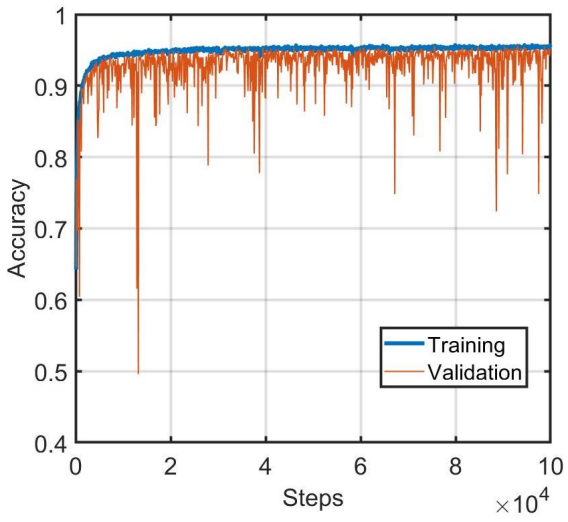


**Figure 4.** Performance of whole-vehicle matching neural network during training.

**Table 3.** Vehicle matching network performance.

| | **Matching Accuracy** |
|---|---|
| Training | 95.45% |
| Validation | 95.20% |

As the threshold is varied, the accuracy value can change; thus, picking an appropriate threshold for this metric is important. In order to evaluate the performance of the vehicle matching network with different threshold values, the true positive rate (TPR), true negative rate (TNR), and matching accuracy were computed as the threshold value was varied for the test set. Accuracy here is defined as the percent of total matches performed (including any number of positive and negative pairs) for which the algorithm produced a correct answer. The resulting curves as functions of threshold value are depicted in Figure 5. From the results, it can be concluded that picking a threshold value between 0.5 and 0.8 results in near-optimal accuracy. It is worth mentioning that some applications may demand a low false negative rate, and others may demand a low false positive rate; thus, it is important to be able to select an appropriate threshold accordingly.

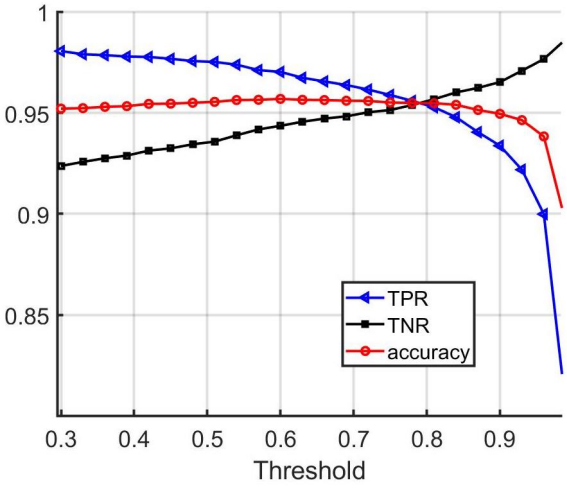


**Figure 5.** Performance of the vehicle matching network on the validation set with different threshold values.

### 5.2. Wheel Locking

It may be of interest to note that the PRIMAVERA dataset consists of vehicle images that have been flipped, rotated, scaled, and shifted such that the front-most and rear-most wheels are always centered on the same two pixel locations in the image. Specifically, an image of size $234 \times 234 \times 3$ was created such that the vehicle is facing to the right (determined by a tracking algorithm applied to the original image sequence) and the rear and front wheels are centered at pixels $[164, 47]$ and $[164, 187]$, respectively. This preprocessing step is here referred to as "wheel locking." The idea is that removing pose variability in the data will make it easier for the neural network to learn true discriminative features. All of the data fusion experiments in this paper used wheel-locked images as input to the whole-vehicle neural network.

An alternative to wheel locking that does not require a wheel detection step is to simply form an input image based on the bounding box returned by the initial vehicle detector algorithm. Due to the variability in how these bounding boxes are constructed, detected instances of a given vehicle may be shifted and scaled relative to one another, and any rotation of the vehicle due to sensor orientation will not be corrected.

To investigate the utility of the wheel-locking preprocessing step, the validation set performance of the whole-vehicle neural network using wheel-locked images as input was compared to when non-wheel-locked images were used as input. In both cases, image intensities were normalized to have pixel values between 0 and 1. A comparison of training and validation set accuracy between the two preprocessing methods is shown in Figures 6 and 7. From the results, it can be concluded that locking the wheels location across the images enhances the matching accuracy.

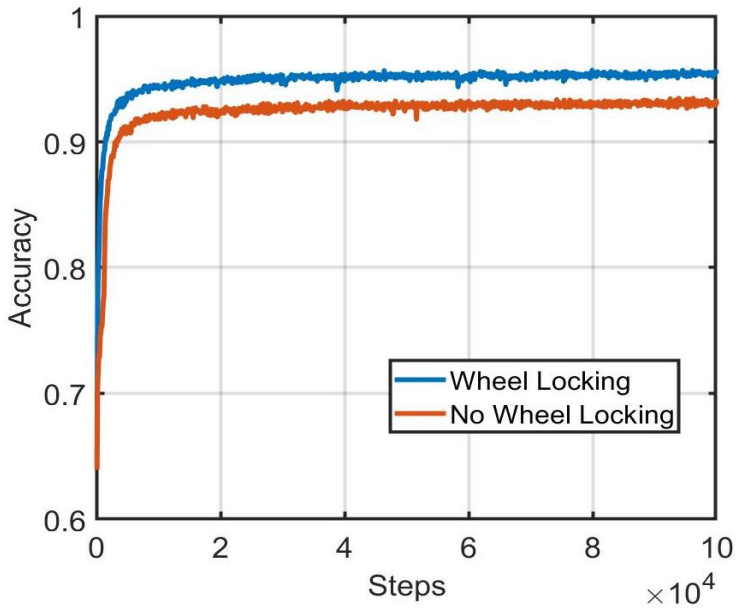


**Figure 6.** Comparison of training set performance during training between wheel-locking and non-wheel-locking preprocessing approaches.

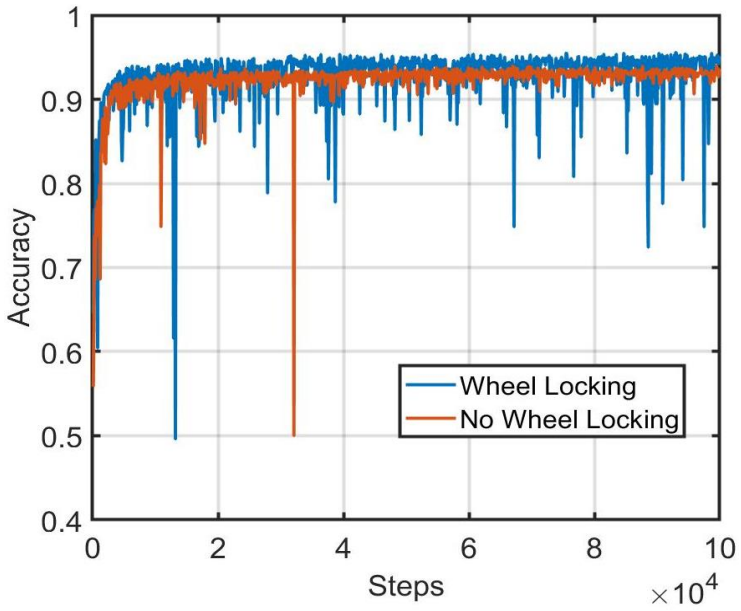


**Figure 7.** Comparison of validation set performance during training between wheel-locking and non-wheel-locking preprocessing approaches.

### *5.3. Wheel Matching Results*

In this subsection, the results of the wheel matching network are shown. The wheel detector described in Section 4.3 was employed to estimate the bounding boxes of wheels' hubcaps from each image. The cropped wheels and their labels were used to train a Siamese network. Specifically, the network was trained using a dataset that contains 228,120 wheel images representing 11,406 unique vehicles. The dataset set was split into two sets: training (159,684 examples) and validation (68,436 examples).

All the wheel images were resized to $100 \times 100 \times 3$. Similarly to the vehicle matching network described in Section 5.1, the wheel matching network consists of two identical branches. Each branch has four layers. The structure of each branch is tabulated in Table 4. The last layer of the Siamese network is a dense layer with 4096 neurons and a sigmoid activation function. The outputs of the two branches are then subtracted from each other and fed into the last layer, which generates the matching score. The predicted matching score is a similarity measure between 0 and 1.

**Table 4.** Wheel matching network structure.

| Layer | Structure | Activation |
|---|---|---|
| Layer 1 | $64 \times 10 \times 10$ | Relu |
| Layer 2 | $128 \times 7 \times 7$ | Relu |
| Layer 3 | $128 \times 4 \times 4$ | Relu |
| Layer 4 | $256 \times 4 \times 4$ | Relu |
| Layer 5 | $4096 \times 1$ | Sigmoid |

The network was trained for 40,000 steps with a learning rate of 0.002. The same dataset described in Section 5.1 was used. After the wheel similarity score is computed for any pair of wheels taken from each image, a threshold is applied to the score. If the average of the two wheel matching scores is more than 0.5, then the two vehicles are declared to be the same, and vice versa. The results for the training and validation are listed in Table 5. By comparing the performances of the vehicle and wheel matching networks in Tables 3 and 5, it can be inferred that the vehicle matching network is more reliable and accurate than the wheel matching network. This was expected because the the vehicle matching network considers the entire image and examines every aspect of the vehicle, whereas many unique vehicles may share identical or very similar wheel designs. However, it is shown later that

combining the results from the vehicle and wheel networks enhances the overall matching performance.

**Table 5.** Wheel matching network performance.

| | **Matching Accuracy** |
|---|---|
| Training | 96.95% |
| Validation | 93.21% |

In Figure 8, the true positive rate (TPR), true negative rate (TNR), and accuracy are shown for different thresholds.

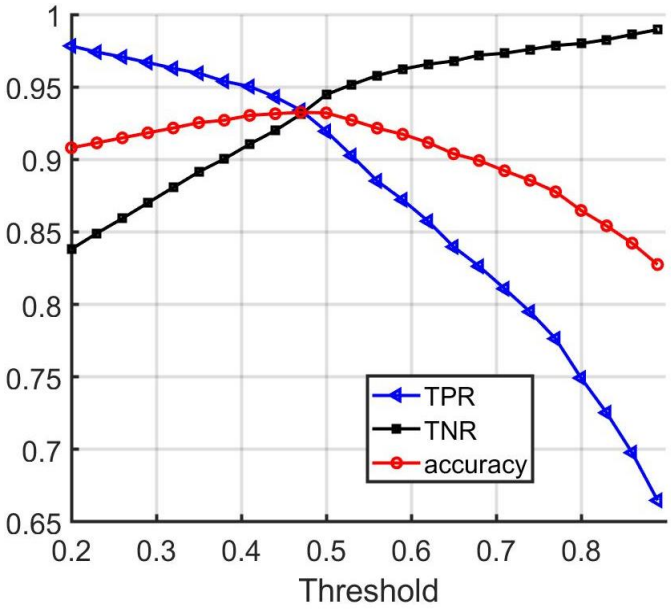


**Figure 8.** Performance of wheel matching network.

From the figure, it can be seen that using a threshold value equal to 0.47 leads to an equal compromise between the error rates.

### *5.4. Decision Fusion Network and Results*

In the following, the results of our decision fusion approach are shown. In this approach, first, all pairs of images go through the whole-vehicle matching network, which generates a whole-vehicle similarity score. Afterwards, the wheel detector described in Section 4.3 is applied to locate and crop the wheels. After at least two wheels are detected from each image, the front wheels of the two images are compared using the wheel matching network. The back wheels are also matched between the two images. As a result, three similarity scores are generated: one whole-vehicle score and two wheel scores. These steps are repeated for all the pairs of vehicles in the dataset.

As a baseline for decision fusion, the average of the three matching scores is computed. Taking the average of the scores has a reasonable chance of providing good results and is the basis of comparison for our deep fusion approach. In Figure 9, the baseline performance is evaluated by varying the threshold, similarly to the experiments done in Section 5.1. In addition, the receiver operating characteristic curves, or ROC curves, are shown in Figure 10. In Table 6, the testing matching accuracy is shown after applying a threshold value equals to 0.5 and utilizing the vehicle matching score, the wheel matching score, or both.

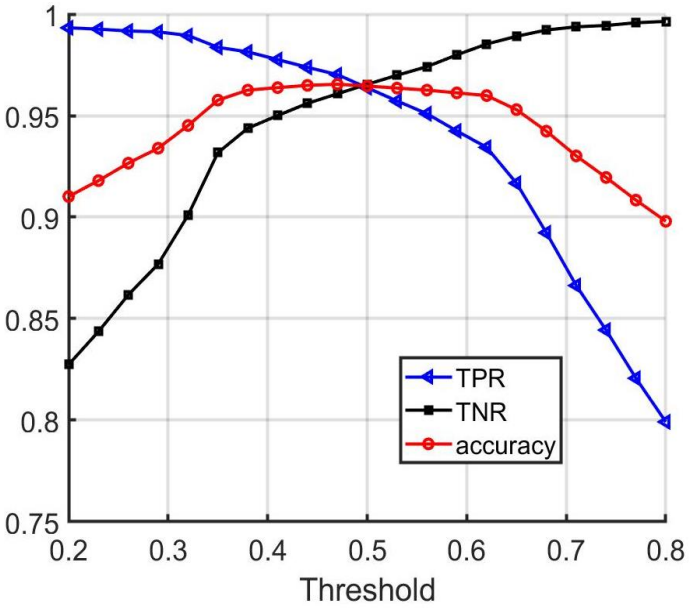


**Figure 9.** Performance of decision fusion by averaging.

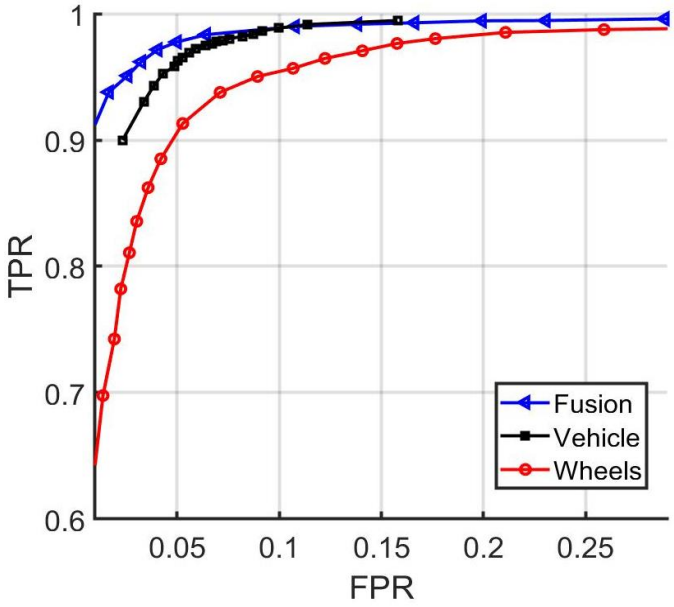


**Figure 10.** ROC curves comparing the performances of whole-vehicle-only matching, wheels-only matching, and averaging-based decision fusion of the two matching approaches.

In addition, in Figures 11–13 a comparison is provided between the baseline, vehicle, and wheel matching accuracy, true positive rate, and true negative rate for distinct threshold values. From the results, it can be concluded that combining the decisions from the vehicle and wheel matching network enhances the matching performance at some operating thresholds. The baseline approach leverages the complementary information from the wheels and provides a more accurate performance in comparison to using the vehicle images alone.

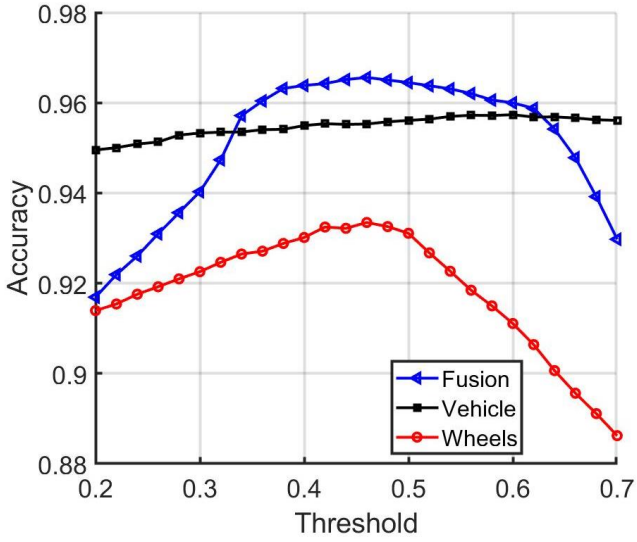


**Figure 11.** Comparison of the baseline, vehicle, and wheel-network-matching accuracies.

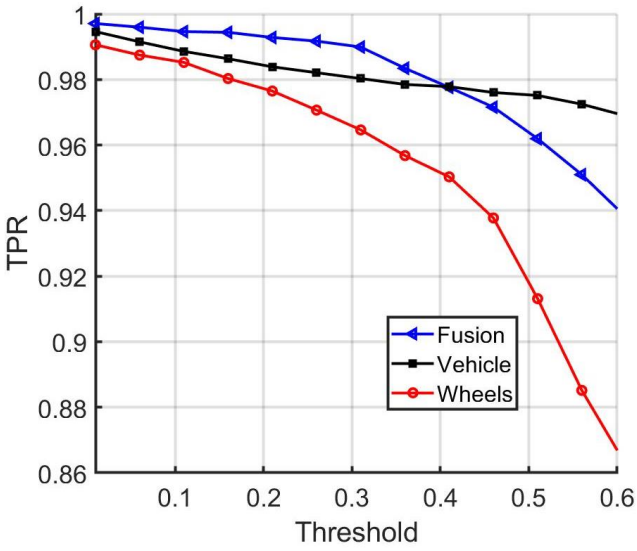


**Figure 12.** Comparison of the baseline, vehicle, and wheel-network-matching true positive rates.

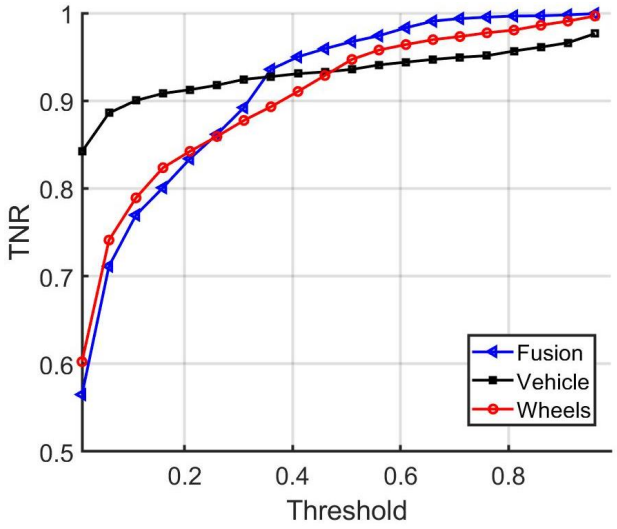


**Figure 13.** Comparison of the baseline, vehicle, and wheel-network-matching true negative rates.

**Table 6.** Average fusion network performance on test data.

| | **Matching Accuracy** |
|---|---|
| Vehicle Matching Score | 95.5% |
| Wheel Matching Scores | 93.21% |
| Average | 97.63% |

In order to substantiate our reasoning, some illustrative matching instances are shown in Figures 14–16 that highlight the utility of the wheel matching. Although each pair of vehicles represents a negative match, i.e., two different vehicles, the whole-vehicle similarity scores are above the threshold, which, by themselves, indicates that they are positive matches. On the other hand, the wheel similarity scores are below the threshold, which suggests a negative match. These examples show the utility of including wheel-specific similarities along with the whole-vehicle similarity to improve matching accuracy.

To better explore the potential contribution of decision fusion in this scenario, a deep decision fusion network was constructed. Using both the vehicle and the wheel similarity scores, a multi-layered, fully connected network was trained that combines the scores and provides a final decision. This can be seen as combining the three scores by performing a smart weighted averaging of the decisions from the wheel and vehicle matching networks. Figure 17 shows the high-level block diagram for the implementation of the deep fusion network. This network is composed of four fully connected layers. The structure of the network is depicted in Table 7. The inputs to the network are the three similarity scores, and the output is the fused similarity score.

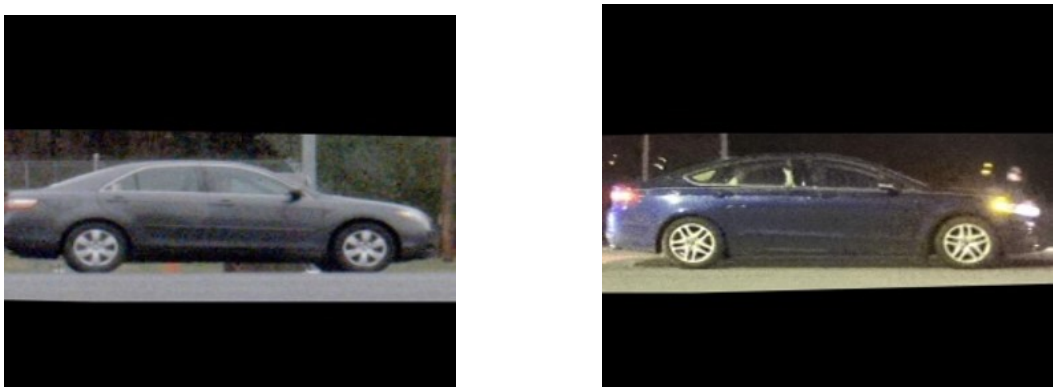

**Figure 14.** Vehicle similarity score = 0.958; wheel similarity scores = 0.01, 0.001.

**Figure 15.** Vehicle similarity score = 0.64; wheel similarity scores = 0.02, 0.01.

**Figure 16.** Vehicle similarity score = 0.91; wheel similarity scores = 0.01, 0.03.

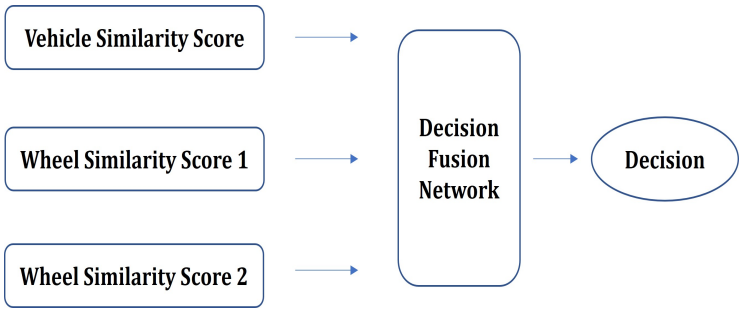


**Figure 17.** Decision fusion network.

**Table 7.** Decision fusion network structure.

| Layer | Structure | Activation |
|---|---|---|
| Layer 1 | 100 | Relu |
| Layer 2 | 70 | Relu |
| Layer 3 | 20 | Relu |
| Layer 4 | 1 | Sigmoid |

Matching scores from the original validation set were split into training (70%) and testing (30%). The new training set had 6265 pairs of vehicles, and the new testing set had 2685 pairs. The network was trained for 500 epochs. The performance of the deep fusion network was tested on the new testing set that contained 2685 pairs of vehicles. The training accuracy was 98.47%. The performance of the decision fusion network was subsequently compared to the baseline, soft voting, and majority voting performances on the testing set. The 95% confidence interval was also calculated, and the results are depicted in Table 8. This comparison shows the advantage of using a decision fusion deep learning method to smartly combine the three similarity scores as opposed to simply averaging the scores or performing soft and hard voting.

**Table 8.** Comparison between fusion methods and no fusion on a portion of the testing data.

| | Fusion Network | Soft Voting | Baseline | Majority Voting | Vehicle Score | Wheel Scores |
|---|---|---|---|---|---|---|
| Accuracy | 97.77 ± 0.56% | 97.28 ± 0.62% | 96.31 ± 0.71% | 95.68 ± 0.77% | 95.46 ± 0.79% | 92.93 ± 0.97% |

In Figure 18, a comparison among the fusion network, majority voting, and baseline matching accuracy is shown for distinct threshold values. In addition, the ROC Curves are shown in Figure 19.

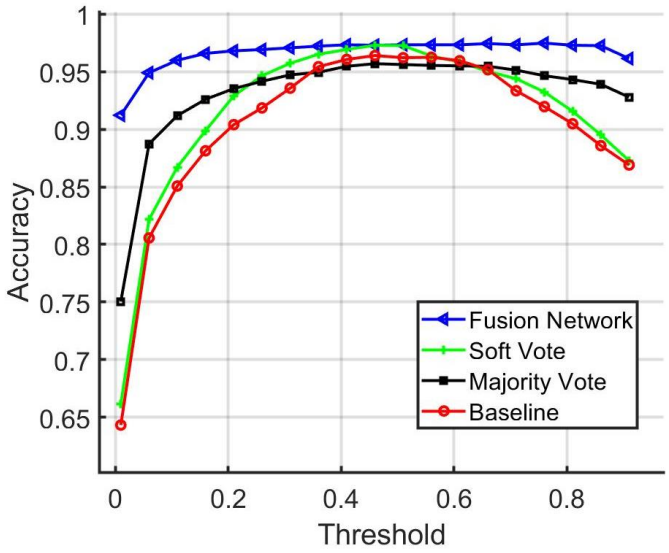


**Figure 18.** Comparison of baseline, majority vote, soft vote, and fusion network matching accuracy.

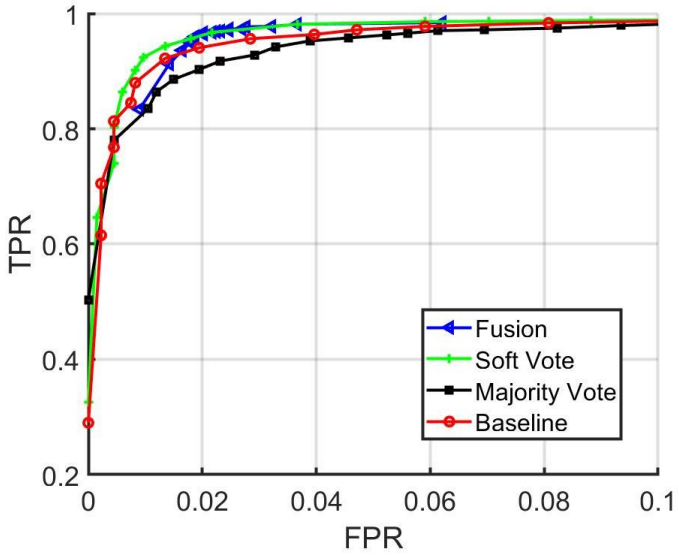


**Figure 19.** Baseline, majority vote, soft vote, and fusion network ROC curves.

## 6. Conclusions and Future Work

In this paper, a deep neural network approach was proposed for decision fusion to match pairs of vehicles through recovering feature signatures from vehicle imagery using Siamese networks. After training the vehicle and the wheel matching networks, the learned networks were then utilized to match new observed data. It was shown that leveraging pattern information specific to the wheels provided supplementary information about the vehicles in question. Experimental results showed a significant improvement in the matching accuracy after fusing the similarity scores from pairs of vehicles and their corresponding wheels' signatures. In addition, our model performed well under diverse illumination conditions. Proposed future work includes investigating the use of vehicle imagery collected by drones for vehicle matching. Drone images may be able to provide supplementary information about the vehicles in question; for example, two similar vehicles could potentially be distinguished if their angled or elevated views reveal distinguishing differences. Moreover, on-demand data acquisition by drones would be more flexible than other traditional methods. In addition, additional sources of information could be fused, such as those describing other distinctive regions of the vehicle.

**Author Contributions:** Conceptualization, S.G., R.A.K., and R.T.; methodology, S.G., R.A.K., and R.T.; software, S.G., R.A.K., and R.T.; validation, S.G., R.A.K., and R.T.; formal analysis, S.G., R.A.K., and R.T.; investigation, S.G., R.A.K., and R.T.; resources, R.A.K.; data curation, S.G., R.A.K., and R.T.; writing—original draft preparation, S.G.; writing—review and editing, R.A.K.; visualization, S.G.; supervision, R.A.K.; project administration, R.A.K.; funding acquisition, R.A.K. All authors have read and agreed to the published version of the manuscript.

**Funding:** This manuscript has been authored by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the US Department of Energy (DOE). By accepting the article for publication, the publisher acknowledges that the US government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for US government purposes. DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (https://www.energy.gov/downloads/doe-public-access-plan accessed on 1 June 2021).

**Institutional Review Board Statement:** Ethical review and approval were waived for this study after the Oak Ridge Site-wide Institutional Review Board (ORSIRB) reviewed the study and determined that the proposed work is not human subjects research.

**Informed Consent Statement:** Because this study was determined not to be human subjects research, informed consent was not applicable.

**Data Availability Statement:** Kerekes, R.A.; Profile Images and Annotations for Vehicle Reidentification Algorithms (PRIMAVERA). Available online: http://doi.ccs.ornl.gov/ui/doi/367 (accessed on 1 January 2022), doi:10.13139/ORNLNCCS/1841347.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Wang, H.; Sun, S.; Zhou, L.; Guo, L.; Min, X.; Li, C. Local feature-aware siamese matching model for vehicle re-identification. *Appl. Sci.* **2020**, *10*, 2474. [CrossRef]
2. Shen, Y.; Xiao, T.; Li, H.; Yi, S.; Wang, X. Learning deep neural networks for vehicle re-id with visual-spatio-temporal path proposals. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 1900–1909.
3. Wang, Z.; Tang, L.; Liu, X.; Yao, Z.; Yi, S.; Shao, J.; Yan, J.; Wang, S.; Li, H.; Wang, X. Orientation invariant feature embedding and spatial temporal regularization for vehicle re-identification. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 379–387.
4. He, B.; Li, J.; Zhao, Y.; Tian, Y. Part-regularized near-duplicate vehicle re-identification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 3997–4005.
5. de Oliveira, I.O.; Fonseca, K.V.O.; Minetto, R. A two-stream siamese neural network for vehicle re-identification by using non-overlapping cameras. In Proceedings of the 2019 IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan, 22–25 September 2019; pp. 669–673.
6. Wei, X.-S.; Zhang, C.L.; Liu, L.; Shen, C.; Wu, J. Coarse-to-fine: A RNN-based hierarchical attention model for vehicle re-identification. In *Asian Conference on Computer Vision*; Springer: Cham, Switzeland, 2018; pp. 575–591.
7. Wang, H.; Skau, E.; Krim, H.; Cervone, G. Fusing heterogeneous data: A case for remote sensing and social media. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 6956–6968. [CrossRef]
8. Ghanem, S.; Panahi, A.; Krim, H.; Kerekes, R.A.; Mattingly, J. Information subspace-based fusion for vehicle classification. In Proceedings of the 2018 26th European Signal Processing Conference (EUSIPCO), Rome, Italy, 3–7 September 2018; pp. 1612–1616.
9. Ghanem, S.; Roheda, S.; Krim, H. Latent Code-Based Fusion: A Volterra Neural Network Approach. *arXiv* **2021**, arXiv:2104.04829.
10. Xu, L.; Krzyzak, A.; Suen, C.Y. Methods of combining multiple classifiers and their applications to handwriting recognition. *IEEE Trans. Syst. Man, Cybern.* **1992**, *22*, 418–435. [CrossRef]
11. Hellwich, O.; Wiedemann, C. Object extraction from high-resolution multisensor image data. In Proceedings of the Third International Conference Fusion of Earth Data, Sophia Antipolis, France, 26–28 January 2000; Volume 115.
12. Hall, D.L.; Llinas, J. An introduction to multisensor data fusion. *Proc. IEEE* **1997**, *85*, 6–23. [CrossRef]
13. Khaleghi, B.; Khamis, A.; Karray, F.O.; Razavi, N.S. Multisensor data fusion: A review of the state-of-the-art. *Inf. Fusion* **2013**, *14*, 28–44. [CrossRef]
14. Lippman, D. *Math in Society*; David Lippman: Washington, DC, USA, 2017.
15. Dempster, A.P. A generalization of Bayesian inference. *J. R. Stat. Soc. Ser. B (Methodol.)* **1968**, *30*, 205–232. [CrossRef]
16. Shafer, G. *A Mathematical Theory of Evidence*; Princeton University Press: Princeton, NJ, USA, 1976.
17. Ben Atitallah, S.; Driss, M.; Boulila, W.; Koubaa, A.; Ghézala, H.B. Fusion of convolutional neural networks based on Dempster–Shafer theory for automatic pneumonia detection from chest X-ray images. *Int. J. Imaging Syst. Technol.* **2021**, *32*, 658–672. [CrossRef]

18. Buede, D.M.; Girardi, P. A target identification comparison of Bayesian and Dempster-Shafer multisensor fusion. *IEEE Trans. Syst. Man Cybern.-Part A Syst. Hum.* **1997**, *27*, 569–577. [CrossRef]
19. Hou, T.; Wang, S.; Qin, H. Vehicle matching and recognition under large variations of pose and illumination. In Proceedings of the 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, Miami, FL, USA, 20–25 September 2009; pp. 24–29.
20. Jelača, V.; Pižurica, A.; Niño-Castañeda, J.O.; Frias-Velazquez, A.; Philips, W. Vehicle matching in smart camera networks using image projection profiles at multiple instances. *Image Vis. Comput.* **2013**, *31*, 673–685. [CrossRef]
21. Zeng, N.; Crisman, J.D. Vehicle matching using color. In Proceedings of the Conference on Intelligent Transportation Systems, Boston, MA, USA, 12 November 1997; pp. 206–211.
22. Kerekes, R.A. Profile Images and Annotations for Vehicle Reidentification Algorithms (PRIMAVERA). Available online: http://doi.ccs.ornl.gov/ui/doi/367 (accessed on 1 January 2022). [CrossRef]
23. Chicco, D. Siamese neural networks: An overview. In *Artificial Neural Networks*; Springer, 2021;pp. 73–94.
24. Bromley, J.; Bentz, J.W.; Bottou, L.; Guyon, I.; LeCun, Y.; Moore, C.; Säckinger, E.; Shah, R. Signature verification using a siamese time delay neural network. *Int. J. Pattern Recognit. Artif. Intell.* **1993**, *7*, 669–688. [CrossRef]
25. Tzutalin LabelImg. Free Software: MIT License. 2015. Available online: http://github.com/tzutalin/labelImg (accessed on 1 July 2021).
26. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
27. Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; Springer: Cham, Switzerland, 2014; pp. 740–755.
28. Kingma, D.P.; Ba, J. Adam: A Method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.