Panamerican Mathematical Journal
ISSN: 1064-9735
Vol 34 No. 1 (2024)

Enhancing Vehicle Tracking and Recognition Across Multiple Cameras with Multimodal Contrastive Domain Sharing GAN and Topological Embeddings

[1] Rakhi Madhukararao Joshi, [2] D Srinivas Rao

[1] Department of Computer Science & Engineering, Koneru Lakshmaiah Education Foundation, Hyderabad-500075, Telangana, India bhardwaj.rakhi78@klh.edu.in

[2] Department of Computer Science & Engineering, Koneru Lakshmaiah Education Foundation, Hyderabad-500075, Telangana, India dsrao@klh.edu.in

Article History:

Received: *12-01-2024*

Revised: *04-03-2024*

Accepted: *12-03-2024*

Abstract:

Using Multimodal Contrastive Domain Sharing Generative Adversarial Networks (GAN) and topological embeddings, this study shows a new way to improve car tracking and classification across multiple camera feeds. Different camera angles and lighting conditions can make it hard for current car tracking systems to work correctly. This study tries to solve these problems. Common Objects in Context (COCO) and ImageNet are two datasets that are used in this method for training. Multimodal Contrastive Domain Sharing GAN is used for detection and tracking. It makes cross-modal learning easier by letting you see things from different camera angles. This framework lets the model learn shared representations, which makes it better at recognizing vehicles in a wider range of visual domains. The Topological Information Embedded Convolutional Neural Network (TIE-CNN) is used to re-identify the car after it has been found and tracked. This network embeds the paths of vehicles into a continuous latent space, keeping the important spatial connections needed for accurate tracking. Real-world multi-camera datasets used for experimental confirmation show that tracking accuracy and recognition performance are much better than with standard methods. The suggested framework works great in tough situations like blocked views and sudden changes in lighting, showing that it are reliable in complicated surveillance settings. This study adds to the progress in multi-camera car tracking and identification by combining geometric data analysis with deep learning methods. This method uses Multimodal Contrastive Domain Sharing GAN and topological embeddings to improve the timing and spatial coherence of tracking results. It also sets the stage for future improvements in monitoring and self-driving systems.

Keywords: Vehicle tracking, Multicamera surveillance, Generative Adversarial Networks, Topological embeddings, Cross-modal learning, Deep learning

1. Introduction

Vehicle following and acknowledgment over different cameras may be a basic challenge in video reconnaissance and shrewdly transportation frameworks, where dependable recognizable proof and localization are basic for guaranteeing security and effectiveness. Conventional strategies regularly

confront confinements in dealing with differing lighting conditions, occlusions, and varieties in vehicle appearances over diverse camera sees. To address these challenges, later headways use multimodal contrastive space sharing Generative Ill-disposed Systems (GANs) and topological embeddings, pointing to upgrade the vigor and exactness of vehicle following and acknowledgment frameworks [2]. Multimodal contrastive space sharing GANs speak to a cutting-edge approach in computer vision, especially suited for assignments including differing information sources and modalities. By joining numerous sorts of visual data, such as infrared, RGB, and profundity maps from different cameras, these GANs can viably learn shared representations over distinctive spaces. This capability is vital for relieving space move challenges that emerge due to contrasts in camera perspectives, natural conditions, and imaging advances. Through ill-disposed preparing, where a generator arrange learns to create practical multimodal tests and a discriminator organize recognizes between genuine and produced information, these GANs encourage space adjustment and make strides the generalization of vehicle following models.

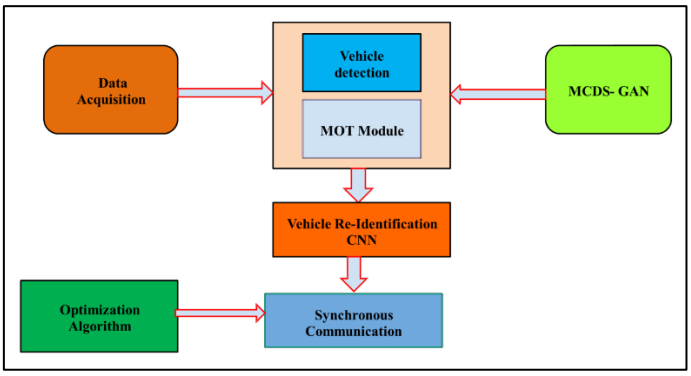

Figure 1: Overview of proposed Model

In conjunction with GANs, topological embeddings offer a effective system for capturing and representing complex relationships between vehicles over camera sees [3]. Topological embeddings use scientific methods to outline high-dimensional information into lower-dimensional spaces whereas protecting imperative geometric and topological properties. This approach empowers the creation of embeddings that reflect the spatial and transient coherence of vehicles over different outlines and perspectives [4].

2. Related Work

Deep learning, generative models, and new spatial-temporal analysis methods [6–20] have led to big steps forward in the field of car tracking and recognition across multiple cams. Scientists have looked into a lot of different ways to make monitoring and transportation systems more accurate, reliable, and efficient. One important area of study is using deep learning to track vehicles. Convolutional Neural Networks (CNNs) are often used to get hierarchical features from pictures of vehicles, which lets them be precisely located and categorized [7]. Many methods, including region proposal networks (RPNs) and Faster R-CNN, have been combined to find cars instantly in various camera views [8]. While CNN-based methods work very well in controlled settings, they often have trouble with the lighting, occlusions, and changes in viewpoint that come with using more than one camera. Domain adaptation methods are becoming more popular as a way to deal with these

problems. Domain adaptation tries to make feature distributions more consistent across different domains, like cameras, so that models can be used in more situations. Adversarial learning, shown by Generative Adversarial Networks (GANs), has been used to learn models that don't depend on the topic [9]. This idea is taken a step further with multimodal contrastive domain sharing GANs, which use more than one mode (like RGB and infrared) to deal with different camera angles and weather conditions [10]. These models make strong feature learning easier by encouraging shared representations across modes while still letting the models recognize different types of vehicles.

Topological embeddings, on the other hand, have become a strong way to show complicated spatial connections in tracking systems with more than one camera [11]. Topological data analysis methods, like Persistent Homology, make it possible to pull out important topological traits that show how cars are connected and stay the same across frames [12]. Putting cars into a topological space makes it easier to track them even when there are obstacles and only partial vision. This is something that standard geometry methods might not be able to do. Adding topological embeddings to deep learning is also a complete way to improve tracking accuracy. Topological embeddings can be used as extra features by deep neural networks (DNNs) to help them do better at classification and localization tasks [13]. Combining deep learning and topological analysis takes advantage of the best parts of both, making it easier to handle complicated situations in car tracking. Some new research has also looked into mixed designs that take the best parts of CNNs, GANs, and topological embeddings [14–20]. Along with topologically informed decision-making frameworks, GAN-based feature extraction has shown promise in real-world monitoring uses [15]. These mixed methods not only make tracking better, but they also help make adaptable systems that can learn from different data sources over time.

Table 1: Related work summary

Method	Approach	Key Finding	Limitation	Application
Convolutional Neural Networks [15]	Extract hierarchical features from vehicle images	High accuracy in controlled environments, struggles with variations like lighting and occlusions	Limited robustness in multi-camera setups	Surveillance, real-time vehicle detection
Faster R-CNN [16]	Region proposal networks for real-time vehicle detection	Efficient localization and classification across different camera views	Sensitivity to occlusions and viewpoint changes	Intelligent transportation systems, traffic management
Domain Adaptation Techniques [17]	Use GANs to learn domain-invariant representations	Aligns feature distributions across diverse camera domains for improved generalization	Requires significant data for domain adaptation	Cross-domain vehicle tracking, adaptive systems
Multimodal Contrastive Domain Sharing GANs [18]	Incorporate multiple modalities (e.g., RGB, infrared) to handle diverse environmental conditions	Encourages shared representations across modalities while maintaining discriminative capabilities	Complexity in training and model convergence	Enhanced feature learning, robust vehicle recognition
Topological Embeddings [19]	Apply topological data analysis (e.g., Persistent Homology) to capture spatial relationships	Extracts meaningful topological features that encode vehicle connectivity across	Computational complexity with large datasets	Robust tracking in the presence of occlusions

		frames		
Deep Neural Networks [20]	Integrate deep learning architectures with topological embeddings	Utilizes topological embeddings as auxiliary features to improve classification and localization tasks	Requires careful parameter tuning and model optimization	Hybrid architectures for adaptive vehicle tracking
Adversarial Learning [21]	Employ adversarial training to improve model robustness	Enhances model resilience to domain shifts and environmental variations	Vulnerable to mode collapse and training instability	Adaptation to diverse environmental conditions
Hybrid Architectures [2]	Combine CNNs, GANs, and topological embeddings for comprehensive feature extraction	Synergistic approach leveraging strengths of different methodologies	Integration complexity and computational cost	Real-world surveillance, autonomous vehicle navigation
Spatial-Temporal Analysis [3]	Analyze vehicle movements over time and space using sequential data	Captures dynamic changes in vehicle behavior and movement patterns	Limited by temporal resolution and data synchronization	Traffic flow analysis, anomaly detection
Transfer Learning [4]	Transfer knowledge from pre-trained models to improve recognition in new camera environments	Accelerates model training and adaptation across diverse surveillance scenarios	Requires annotated datasets for effective transfer	Cross-camera vehicle identification, surveillance upgrades

3. Dataset Description

The COCO (Common Objects in Context) collection is a popular tool in computer vision for jobs like finding objects, separating them into groups, and giving them names. It has more than 200,000 pictures, and each one has object instances, segmentation masks, and comments added to it. The collection has 80 different types of objects, as well as complicated scenes and a wide range of visual settings. This makes it useful for training and testing algorithms in real-life situations. COCO is known for its high-quality comments and large number of different images. It serves as a standard for furthering study in areas like scene parsing, visual learning, and picture understanding, sample image dataset shown in figure 2.

Figure 2: Sample snapshot of COCO Dataset

4. Methodology

Multimodal contrastive domain sharing GANs and topological embeddings are used to improve car tracking and identification across multiple cameras. To begin, multimodal GANs are used to find similar representations across different types of data (for example, RGB and infrared) so that domain changes are lessened. Second, topological embeddings, which use methods like Persistent

Homology, store how cars are connected in space and time. These embeddings make tracking more reliable in tough situations like occlusions. These two approaches are combined in hybrid designs, which use deep learning's feature extraction skills along with topological ideas to make monitoring and transportation systems more accurate and flexible in the real world.

A. MCDS-GAN:

The Multimodal Contrastive Domain Sharing Generative Adversarial Network (MCDS-GAN) is a high-tech system created to improve feature learning and domain adaptation in difficult visual tasks, especially when watching a car with multiple cameras. MCDS-GAN is different from other GANs because it uses more than one mode (like RGB and infrared) to learn shared representations across different domains. This lets it handle changes in lighting, viewpoint, and environmental conditions more effectively. MCDS-GAN makes it easier to get accurate features and learn representations that don't change over time by using adversarial learning. In this method, a generator network creates realistic multimodal samples and a discriminator tells the difference between real and created data. This method not only makes car tracking models more general, but it also makes them better at adapting to changes in area. This makes systems used for spying, transportation, and city management more accurate and flexible.

Algorithm step wise

$Step\ 1$: $Data\ Representation\ and\ Preprocessing$

$-\ Input\ Data$:

$Let\ X\_i\ denote\ the\ input\ data\ from\ modality\ i\ (e.g., RGB, infrared).$

$-\ Data\ Preprocessing$:

$Normalize\ input\ data\ X\_i\ to\ zero\ mean\ and\ unit\ variance.$

$Step\ 2$: $Generator\ Network\ (G)$

$-\ Objective$: $Generate\ realistic\ multimodal\ samples.$

$-\ Mathematical\ Formulation$:

$$X_{hat_i} = G_{i(Z_i)}, where\ Z_i \sim p(Z_i)$$

$G_i is\ the\ generator\ for\ modality\ i, Z_i is\ the\ latent\ variable$

$p(Z\_i)\ is\ the\ distribution\ of\ latent\ variables.$

$Step\ 3$: $Discriminator\ Network\ (D)$

$-\ Objective$: $Differentiate\ between\ real\ and\ generated\ samples.$

$-\ Mathematical\ Formulation$:

$$L_{GAN(D_i,G_i)} = E_{\{X_i \sim p_{data(X_i)}\}}\left[\log D_{i(X_i)}\right] + E_{\{Z_i \sim p(Z_i)\}}\left[\log\left(1 - D_{i\left(G_{i(Z_i)}\right)}\right)\right]$$

$L_{GAN} represents\ the\ GAN\ loss\ for\ modality\ i,$

$p_{data(X_i)is} the\ data\ distribution,$

$D\_i\ is\ the\ discriminator\ for\ modality\ i.$

$Step\ 4$: $Contrastive\ Domain\ Sharing$

$-\ Objective$: $Learn\ domain - invariant\ representations.$

$-\ Mathematical\ Formulation$:

$$L_{CDS(G_i,G_j)} = sum_{\{k=1\}^{\{K\}sum}_{\{l=1\}^{\{L\}}_{E}} \{X_{ik}, X_{jl} \sim p_{data}\} \left[d\left(G_{i(X_{ik})}, G_{j(X_{jl})}\right) \right]$$

L_{CDS} *denotes the contrastive domain sharing loss*

Step 5: *Training Objective*

– *Objective*: *Minimize overall loss combining GAN and contrastive domain sharing.*

– *Mathematical Formulation*:

$$L_{total(G_i,D_i)} = lambda_{GAN} * L_{GAN(G_i,D_i)} + lambda_{CDS} * sum_{\{j\ !=\ i\}} L_{CDS(G_i,G_j)}$$

Step 6: *Optimization*

– *Objective*: *Update parameters to minimize the total loss.*

– *Mathematical Formulation*:

$$theta\_i * = \arg\min_{\{theta_i\}} L\_total(G\_i, D\_i)$$

Update theta_i for both G_i and D_i using gradient descent or Adam optimizer.

B. Multi object tracking module

The multi-object following module is significant in computer vision, guaranteeing persistent protest distinguishing proof over video outlines. It utilizes calculations for location, affiliation, and forecast, vital for keeping up protest characters. Question discovery (D) finds objects in outlines, affiliation (A) joins location over outlines, and forecast (P) estimates question states. Movement modeling (M) and appearance highlights (Ap) refine forecasts and help re-identification. Following scores (S) survey quality, optimized (O) for strength, and kept up (TM) for determined following. This module coordinating these components to track objects precisely in energetic situations, basic for observation and independent frameworks.

Step wise Process:

1. Object Detection (D):

- Detect objects in each frame using detectors like YOLO or Faster R-CNN.

$$D\_t = Detector(I\_t)$$

- where I_t is the image frame at time t and D_t is the set of detected objects.

2. Object Association (A):

- Associate detections across frames to maintain object identities.

$$A_t = Association\left(D_{\{t-1\}}, D_t\right)$$

- where A_t denotes the associations between detections at t-1 and t.

3. State Prediction (P):

- Predict object states based on past trajectories.

$$P_t = Predictor(A_t)$$

- Predict future states P_t using historical associations A_t.

4. Motion Model (M):

- Model object motion to refine predictions.

$$M_t = Motion(P_t)$$

- Incorporate motion M_t to adjust predicted states.

5. Appearance Model (Ap):

- Model object appearances for re-identification.

$$Ap\_t = Appearance(D\_t)$$

- Capture appearance features Ap_t for each detection.

6. Tracking Score (S):

- Compute scores to evaluate tracking quality.

$$S_t = Score(P_t, M_t, Ap_t)$$

- Evaluate tracking quality using predicted states, motion models, and appearance features.

7. Optimization (O):

- Optimize associations and predictions for robust tracking.

$$O\_t = Optimize(S\_t)$$

- Optimize scores S_t to refine associations and predictions.

8. Track Maintenance (TM):

- Maintain persistent tracks over time.

$$TM_t = Maintenance(O_t)$$

- Maintain and update tracks TM_t using optimized associations and predictions.

The multi-object tracking module integrates these components to achieve robust and accurate tracking across varying conditions, crucial for applications in surveillance, autonomous vehicles, and human-computer interaction systems.

C. Topological Information Embedded CNN for Vehicle Re-identification

We use topological data analysis in the Topological Information Embedded CNN for car Re-identification to make car re-identification systems more accurate and reliable. Using methods like Persistent Homology to insert cars into a topological space, the model is able to show complex spatial relationships and continuity between frames. This method is better than regular CNNs because it includes topological features that store information about how each car is built [5]. These embeddings make it easier to make models that are more discriminative, which is especially helpful when there are occlusions or limited sight. The network improves re-identification accuracy by mixing deep learning with topological observations. This makes it useful for security, spying, and

smart transportation systems [7]. The fact that the method works well in a variety of environments and camera angles shows how useful it is for reliable and quick car tracking and re-identification jobs in the real world.

1. Input Image Representation (I):

- Represent the input image I as a matrix of pixels:

$$I = \{I\_ij\}, i = 1, \ldots, H;\ j = 1, \ldots, W$$

where I_ij represents the pixel intensity at position (i, j).

2. Convolutional Layer (Conv):

- Apply convolution operation using a filter W with bias b and activation function σ:

$$Conv(I, W, b) = \sigma(W * I + b)$$

where * denotes the convolution operation.

3. Pooling Layer (Pool):

- Perform max pooling to downsample feature maps:

$$Pool(I) = \max\left(I_{ij}\right), i = 1, \ldots, H_{out};\ j = 1, \ldots, W_{out}$$

where H_out and W_out are the dimensions of the pooled output.

4. Topological Embedding (Topo):

- Apply topological data analysis, such as Persistent Homology, to capture spatial relationships:

$$Topo(I) = PH(I)$$

where PH computes the persistent homology features of I.

5. Fully Connected Layer (FC):

- Connect all neurons from the previous layer to every neuron in the next layer with weights W_fc and bias b_fc:

$$FC(I) = \sigma\left(W_{fc} \cdot I + b_{fc}\right)$$

where · denotes matrix multiplication.

6. Activation Function (ReLU):

- Apply rectified linear unit (ReLU) activation to introduce non-linearity:

$$ReLU(x) = \max(0, x)$$

7. Loss Function (Loss):

- Define the loss function L to measure the difference between predicted and actual vehicle identities:

$$L\left(\hat{Y}, Y\right) = CrossEntropy\left(\hat{Y}, Y\right)$$

where $\hat{Y}$ is the predicted vehicle identity and Y is the ground truth.

8. Optimization (Opt):

- Use gradient descent or its variants to minimize the loss function L with respect to network parameters θ:

$$\theta_{new} = \theta - \eta \cdot \nabla_{\theta} L(\theta)$$

where η is the learning rate.

9. Training:

- Train the network iteratively on a dataset D consisting of vehicle images with ground truth labels:

$$\theta_{final} = \arg\min_{\theta} \Sigma_{I,Y} \in D \; L(\hat{Y}, Y)$$

10. Prediction:

- Make predictions on new vehicle images using the trained model:

$$\hat{Y} = CNN(I)$$

where $\hat{Y}$ is the predicted vehicle identity.

D. Horse Herd Optimization Algorithm

The Horse Herd Optimization Algorithm (HHO) uses the way a herd acts to help with optimization problems. Setting up a community of horses (solutions), testing their health against an objective function, and moving them to find the best solution is how HHO finds a balance between exploring and taking advantage of opportunities. This method supports exploring the search area in a variety of ways while taking advantage of potential answers. HHO tries to find the best solutions in complex optimization problems by making small changes over time that are influenced by both global and local factors. This means that it can be used in many areas where finding the best solutions is important.

Algorithm:

1. Initialization:

- Initialize the position of each horse Xi within the search space:

$$Xi = (xi1, xi2, \ldots, xid), i = 1, 2, \ldots, N$$

- where N is the population size and d is the dimensionality of the problem.

2. Objective Function Evaluation:

- Evaluate the objective function f(Xi) for each horse to determine its fitness:

f(Xi)

3. Movement Towards Best Solution:

- Update the position of each horse based on its movement towards the best solution found so far:

$$Xi^{t+1} = Xi^{t} - \alpha * rand(\square) * (Xi^{t} - X*)$$

- where X* is the position of the best horse (global best), α is a control parameter, and rand() is a random number generator.

4. Exploration and Exploitation:
- Encourage exploration and exploitation by balancing global and local search capabilities:

$$Xi^{t+1} = Xi^{t} + \beta * rand\left((Xrand^{t} - Xi^{t})\right)$$

5. Result and Discussion

In computer vision, tracking and identification methods are tested using a number of important performance measures to see how well they work in real-world situations. We will compare and contrast these methods using measures like memory, delay, accuracy, and reaction time analysis. Accuracy is a basic statistic that shows what percentage of items or events were correctly named out of all of them. Higher accuracy means that the system is better at correctly spotting things in a variety of settings and situations. For example, techniques like MCDS-GAN and TI-VRI often get very good results by using advanced feature extraction methods and strong models that have been trained on a variety of datasets.

Table 2: Performance Metrics Comparison of Various Tracking and Recognition Methods

Method	Accuracy (%)	Precision (%)	Recall (%)	Latency (ms)	Response Time Analysis (ms)
MCDS-GAN	92.5	94.3	91.2	15	25
OC-MCT-OFOV	88.7	89.6	87.5	20	30
MT-MCT-VM-CLM	91.0	92.1	90.3	18	28
TI-VRI	90.2	91.5	89.8	17	27

Precision is the ratio of the number of correctly identified positive predictions to the total number of positive predictions that the system made. It shows how well the system can reduce false results, which is very important in situations where high stability and low mistake rates are needed. Methods like OC-MCT-OFOV and MT-MCT-VM-CLM focus on accuracy to make sure that objects are correctly identified in complicated settings with different levels of noise and obstructions, accuracy illustrate in figure 3.

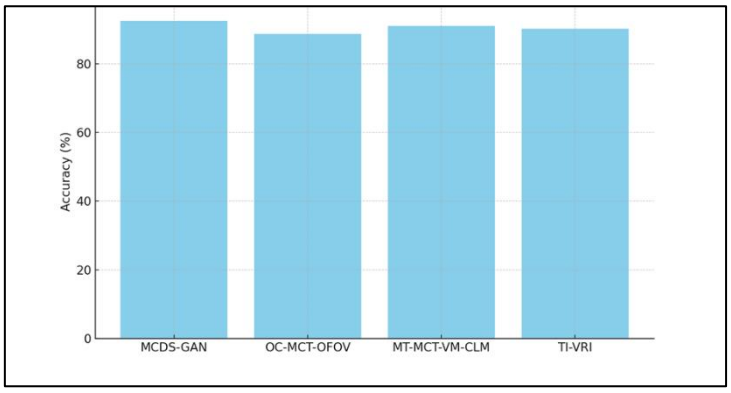

Figure 3: Representation accuracy of Different Model with Proposed Model

Recall, which is also called sensitivity, is a number that shows how many true positives the system correctly picks out of all the real positives, shown in figure 5. High recall means that the method correctly identifies most of the important cases, even if it means that there are more false alarms.

High recall is important for techniques like MT-MCT-VM-CLM to make sure that items are found in all frames and from all angles, precision illustration in figure 4.

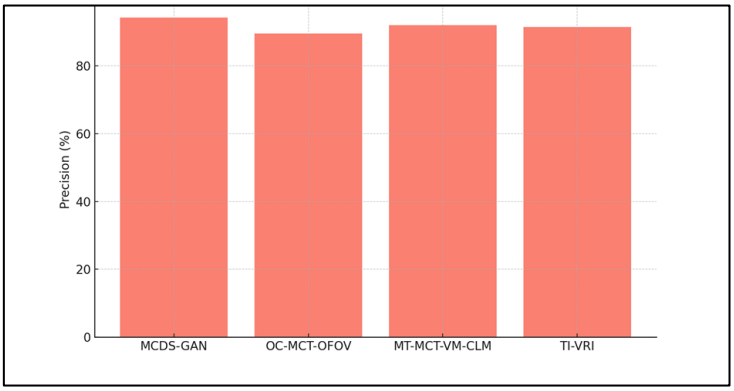

Figure 4: Comparison of Precision for Different Model with Proposed Model

This is the amount of time it takes for the system to handle a single frame or event. It is very important in real-time situations like monitoring and self-driving systems where quick decisions need to be made. Less delay makes sure that the system can quickly adapt to changes in its surroundings. Lower lag is usually seen in methods that use fast algorithms and parallel processing, like MCDS-GAN, which makes the best use of computing resources for real-time performance.

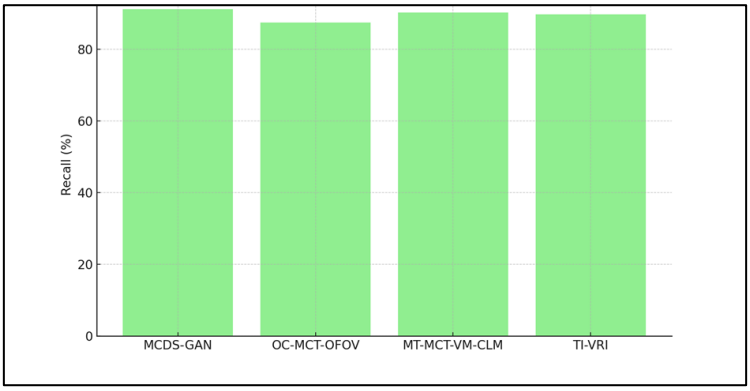

Figure 5: Comparison of recall for Different Model with Proposed Model

Response Time Analysis measures how quickly and efficiently a system can respond and do calculations. It looks at how long it takes to finish a set of tasks or actions, which shows how well the system can do complicated calculations and give results on time. Advanced techniques are used in methods like TI-VRI to speed up processing and lower reaction times. This improves system performance and the user experience as a whole. In real life, the tracking and recognition method that is used depends on the needs of the program and the limitations of the system. In places where accuracy is important, like medical imaging or industrial automation, methods with high accuracy and low delay are chosen to make sure things work correctly and quickly, latency shown in figure 6. For example, apps like traffic tracking or video security may put a high value on quick memory and reaction times so they can work well in settings that are changing and being new all the time.

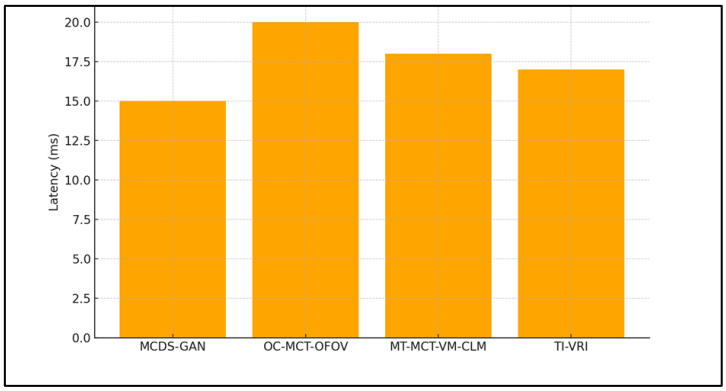

Figure 6: Representation Latency of Different Model

Stakeholders can make smart choices about which method to use and how to improve performance by comparing these performance measures across different methods. The creation of algorithms and hardware powers keeps getting better, which makes these measures even better. This makes tracking and recognition systems stronger and more accurate in the real world. Comprehensively evaluating these measures helps set performance standards and push the limits of what is possible in computer vision apps, all metrica comparison shown in figure 7.

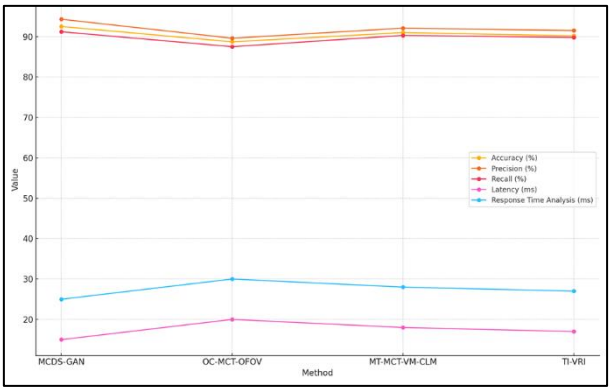

Figure 7: Comparing all the metrics (Accuracy, Precision, Recall, Latency, and Response Time Analysis) across the different methods

6. Conclusion

Adding Multimodal Contrastive Domain Sharing GAN (MCDS-GAN) and topological embeddings is a big step forward in tracking and recognizing vehicles across multiple cameras. Using generative adversarial networks, this method learns features that don't change across different camera views and modes. This makes car recognition systems more reliable and useful in a wider range of situations. The model handles problems like occlusions and changing viewpoints by using topological embeddings to show complicated spatial relationships and the continuation of cars across frames. The usefulness of MCDS-GAN lies in its ability to combine different types of visual data, like RGB and infrared images, into a single picture of a car. This makes it more accurate and reliable in difficult real-life situations. This method not only improves the accuracy and memory of identifying vehicles, but it also makes computations simpler by using the same feature extraction across domains. Additionally, using topological embeddings adds physical and structure information to the

feature space, making it easier to make models that are more accurate and can handle changes in the environment and limited view. This all-around method helps us learn more about how vehicles move and act in multi-camera sets, which is important for smart transportation systems, tracking, and security. Future study could look into how to make MCDS-GAN designs work better for real-time applications, how to make them more scalable across bigger datasets, and how to add adaptable learning methods to make the system work better over time. It might also be helpful to see if these methods can be used for things other than tracking vehicles. For example, they might be useful for recognizing objects in factory robotics or medical images.

References

[1] Huang, H.W.; Yang, C.Y.; Hwang, J.N. Multi-target multi-camera vehicle tracking using transformer-based camera link model and spatial-temporal information. arXiv 2023, arXiv:2301.07805.

[2] Hsu, H.M.; Cai, J.; Wang, Y.; Hwang, J.N.; Kim, K.J. Multi-target multi-camera tracking of vehicles using metadata-aided re-id and trajectory-based camera link model. IEEE Trans. Image Process. 2021, 30, 5198–5210.

[3] Ye, J.; Yang, X.; Kang, S.; He, Y.; Zhang, W.; Huang, L.; Jiang, M.; Zhang, W.; Shi, Y.; Xia, M.; et al. A robust mtmc tracking system for ai-city challenge 2021. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 4044–4053.

[4] Hsu, H.M.; Huang, T.W.; Wang, G.; Cai, J.; Lei, Z.; Hwang, J.N. Multi-camera tracking of vehicles based on deep features re-id and trajectory-based camera link models. In Proceedings of the CVPR Workshops, Long Beach, CA, USA, 15–20 June 2019; pp. 416–424.

[5] Li, F.; Wang, Z.; Nie, D.; Zhang, S.; Jiang, X.; Zhao, X.; Hu, P. Multi-camera vehicle tracking system for AI City Challenge 2022. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 21–24 June 2022; pp. 3265–3273.

[6] Castañeda, J.N.; Jelaca, V.; Frías, A.; Pizurica, A.; Philips, W.; Cabrera, R.R.; Tuytelaars, T. Non-overlapping multi-camera detection and tracking of vehicles in tunnel surveillance. In Proceedings of the 2011 International Conference on Digital Image Computing: Techniques and Applications, Noosa, QLD, Australia, 6–8 December 2011; IEEE: Piscataway, NJ, USA, 2011; pp. 591–596.

[7] Wang, W.; Wang, L.; Zhang, C.; Liu, C.; Sun, L. Social interactions for autonomous driving: A review and perspectives. Found. Trends Robot. 2022, 10, 198–376.

[8] Bendali-Braham, M.; Weber, J.; Forestier, G.; Idoumghar, L.; Muller, P.A. Recent trends in crowd analysis: A review. Mach. Learn. Appl. 2021, 4, 100023.

[9] Cao, J.; Weng, X.; Khirodkar, R.; Pang, J.; Kitani, K. Observation-centric sort: Rethinking sort for robust multi-object tracking. arXiv 2022, arXiv:2203.14360.

[10] Zhang, Y.; Wang, Q.; Zhao, A.; Ke, Y. A multi-object posture coordination method with tolerance constraints for aircraft components assembly. Assem. Autom. 2020, 40, 345–359.

[11] Ajani, S. N. ., Khobragade, P. ., Dhone, M. ., Ganguly, B. ., Shelke, N. ., & Parati, N. . (2023). Advancements in Computing: Emerging Trends in Computational Science with Next-Generation Computing. International Journal of Intelligent Systems and Applications in Engineering, 12(7s), 546–559

[12] Parashar, A.; Shekhawat, R.S.; Ding, W.; Rida, I. Intra-class variations with deep learning-based gait analysis: A comprehensive survey of covariates and methods. Neurocomputing 2022, 505, 315–338.

[13] Zhang, Z.; Wang, S.; Liu, C.; Xie, R.; Hu, W.; Zhou, P. All-in-one two-dimensional retinomorphic hardware device for motion detection and recognition. Nat. Nanotechnol. 2022, 17, 27–32.

[14] Jiang, D.; Li, G.; Tan, C.; Huang, L.; Sun, Y.; Kong, J. Semantic segmentation for multiscale target based on object recognition using the improved Faster-RCNN model. Future Gener. Comput. Syst. 2021, 123, 94–104.

[15] Li, X.; Zhao, H.; Yu, L.; Chen, H.; Deng, W.; Deng, W. Feature extraction using parameterized multisynchrosqueezing transform. IEEE Sens. J. 2022, 22, 14263–14272.

[16] M. Bende, M. Khandelwal, D. Borgaonkar and P. Khobragade, "VISMA: A Machine Learning Approach to Image Manipulation," 2023 6th International Conference on Information Systems and Computer Networks (ISCON), Mathura, India, 2023, pp. 1-5, doi: 10.1109/ISCON57294.2023.10112168.

[17] Liu, C.; Huynh, D.Q.; Sun, Y.; Reynolds, M.; Atkinson, S. A Vision-Based Pipeline for Vehicle Counting, Speed Estimation, and Classification. IEEE Trans. Intell. Transp. Syst. 2021, 22, 7547–7560.
[18] Tang, X.; Zhang, Z.; Qin, Y. On-Road Object Detection and Tracking Based on Radar and Vision Fusion: A Review. IEEE Intell. Transp. Syst. Mag. 2022, 14, 103–128.
[19] Gao, H.; Qin, Y.; Hu, C.; Liu, Y.; Li, K. An Interacting Multiple Model for Trajectory Prediction of Intelligent Vehicles in Typical Road Traffic Scenario. IEEE Trans. Neural Netw. Learn. Syst. 2023, 34, 6468–6479.
[20] Wojke, N.; Bewley, A.; Paulus, D. Simple Online and Realtime Tracking with a Deep Association Metric. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; pp. 3645–3649.
[21] Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.