2023 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops) | 978-1-6654-5381-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/PERCOMWORKSHOPS56833.2023.10150396

PADAAV: Enhancing Perception Systems using GAN-generated Adversarial Augmented Domains for Autonomous Vehicles

Oshin Rawlley, Shashank Gupta
Department of Computer Science & Information Systems
Birla Institute of Technology and Science, Pilani, Rajasthan, India
p20200063@pilani.bits-pilani.ac.in, shashank.gupta@pilani.bits-pilani.ac.in

***Abstract*—In the field of autonomous vehicles (AV), it is crucial for the perceptual systems of the AVs to learn inter-domain adaptations in the absence of paired examples for detecting vehicular instances in unstructured real-world scenarios. One straightforward approach is to train the models directly on labeled synthetic datasets. However, this approach usually fails to achieve generality, owing to the domain bias between the real and fake databases of images. We therefore, propose a novel architecture that produces synthetic images based on cycle consistency, in the absence of labeled pair of images. We test the object detectors Detectron and SSD on four types of curated benchmark datasets to evaluate their robustness in detecting objects such as cars, bikes, and pedestrians on road. The four benchmark datasets contain a diversified set of image corruptions and a few variations are built using the proposed framework such as weather variations. The four datasets are PASCAL VOC and SYNTHIA dataset images, weather-translated images, variation-augmented images, and stylized renderings using Adain style transfer. After conducting extensive investigations, we observe a decreased classification loss when exposed to variable image quality. We also witness that augmenting the training datasets with variations in the wild aided in boosting the generalizing capability of the object detectors. The boost in performance is testified by the testing results showing better mAP**

***Index Terms*—Perception systems, Autonomous Vehicles, Object detection, Generative adversarial networks, inter-domain adaptation.**

I. Introduction

Object detection is an integral task of the advanced driver assistance systems (ADAS) which detects vehicular instances on the roads in real-time [1]. The recent boom in deeplearning methods has largely improved the performance of state-of-the-art object detection models. However, many of the deep learning methods greatly rely on the labeled databases of images that are data-driven. This restricts the capability of the deep learning techniques to generalize to new or unseen real-world road scenarios. The new surroundings (target domain) on the road include all kinds of varying scenarios, fluctuating illumination conditions, and weather distinctions. For example, datasets like KITTI, and Cityscapes, are curated in distinct weather conditions ignoring actual real-world problems like fog, rain, smog, etc. LISA 2010 dataset includes three sequences of AV's video [2]. Urban Traffic Surveillance (UTS) dataset offers camera-taken images of vehicles. SYNTHIA dataset is a synthetic database of images providing fake driving scenes with domains such as morning, evening, dusk, and night [3]. For more genuine artificial images, the GTA dataset comprises images in weather such as night, snow, day, rain, and sunset conditions [4]. These prevailing datasets lack wide-ranging variations in images that are integral for robust object detection in the perception systems of AV [5]. These new surroundings diverge highly from the training set which is used to train the object detection models [6]. Therefore, Generative Adversarial Network (GANs) addresses the domain change issues by generating synthetic instances of real data. The GANs coined by Goodfellow et al. [7] comprises the generator which tries to mimic the real images to create fake images similar to the real ones and the discriminator tries to discriminate the fake images from the real image [8]. While GAN frameworks have shown a substantial amount of success, but they need invertible inter-domain mapping and also to cross-check the accuracy of the produced image with the original one. Our proposed framework has a strict similarity function underlying the source domain and the target domain to validate the same [9]. We propose a new architecture showcasing two types of synthetic weather translations. This synthetic data is then augmented with various style renderings and other augmentations to make a diversified training set on which we test two categories of object detection models: SSD (one-stage) and Detectron (two-staged). Hence, we highlight our major contributions in the manuscript as follows:

- We suggest a novel model for producing fake images under different domains for aiding the object detectors to learn the adverse variations in

Authorized licensed use limited to: BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE. Downloaded on December 24,2024 at 00:52:26 UTC from IEEE Xplore. Restrictions apply.

the wild.

- We designed a combined dataset comprising:two techniques of data augmentations, two renderings of AdaIN fast style transfer [7], and two domain shifts of different weather modelings to cater the adverse data insufficiency and further conducted extensive performance evaluations and statistical analysis to validate the suitability of our combined dataset in the intra-domain sphere. .

II. Proposed Framework

Earlier state-of-the-art object detection models were not able to produce synthetic instances of images in the absence of paired examples of images. In addition, high-level semantic information of the images such as preserving the key features should also be done during the style transfer [10]. Further, the synthetic image should be nearly comparable with the original image that the object detection models are able to make no distinction between them. Therefore, we propose a novel cycle-consistency-based model exhibiting an adversarial behavior for constructing the synthetic image from an input sample [11]. This generated fake image is rebuilt back to the exact input source which makes the fake image indistinguishable from the real one. The fulfillment of this condition aids in providing diversification in the training dataset of the images. Once the fake image is generated by the proposed algorithm, it is essential to validate the closeness of the generated image with the real image. In other words, how much similar the proposed model-produced synthetic image is to the actual input image. The *Algorithm 2* explains the verification of the accurate image transformation process. This translation of one image domain to another domain is usually termed image-to-image translation (UNIT).

A. Problem Definition and Formulation

Suppose there are two unlabeled sets of images of different domains P and Q respectively. The main aim of the proposed architecture is to employ a generator $G_X : P \rightarrow Q$. The G_X maps the image p from P domain i.e., $p \in P$ to image q from Q domain i.e., $q \in Q$. The second generator is employed for training an inverse generator G_Y: $Q \rightarrow P$. This aids in gaining back the synthetic image to the real input one. Therefore, our proposed architecture utilizes two generators for back translation. This is also called cycle consistency. The generator G_X generates the synthetic image and the Discriminator D_X discriminates between the synthetic image of G_X and the real images of the domain Q. Correspondingly, the second generator G_Y and discriminator D_Y functions the same. The overall architecture of the proposed system model is illustrated in *Figure 1* which shows that image $p \in P$ is transformed to Q domain with the help of G_x. To evaluate how well the translation of $G_X(p, z)$ fits into Q, discriminator D_X is used, where, z and z' are some random noise. $G_X(p, z)$ is then reverted to domain P using G_Y resulting in reconstructed image of p i.e $G_Y(G_X(p, z), z')$. In a similar way, $q \in Q$ is translated to P as $G_Y(q, z')$ and then restored as $G_X(G_Y(q, z), z')$. The generators G_X, G_Y are trained to produce synthetic images to fool the discriminators D_X, D_Y and in addition, minimizing the two reconstruction losses:
$||G_X(G_Y(q, z'), z) - q||$ and $||G_Y(G_X(p, z'), z) - p||$.

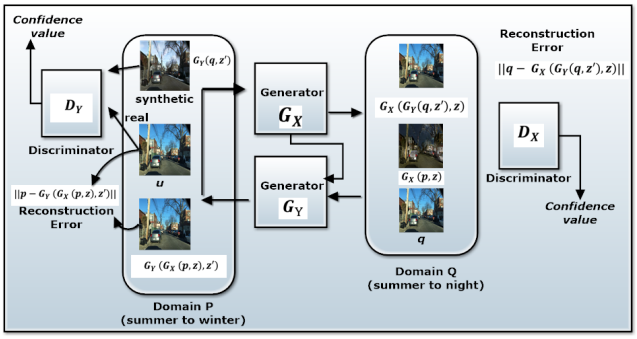

Figure 1: Proposed Network architecture

B. System Architecture

For enhancing the DNN's detection prowess, we consider Cycle-GAN as our reference model to build this proposed architecture [12]. These synthetic instances should be seamlessly in-differentiable from the real image instances. Therefore, our architecture consists of two generators, G_X and G_Y, and a discriminator D that are jointly employed in this adversarial model as shown in *Figure 1*. In addition, we take two object detection models Detectron and SSD to test their detection capability under unseen scenarios generated by the proposed framework. The discriminator outputs the probability of the source image being tested from the training set, say X.

C. Network Configuration

We assumed two identical networks of both the generators G_X and G_Y. The generators are equipped with an equivalent number of up-sampling and down-sampling layers. We do not introduce skip connections between the layers as it disables to transfer high-level of image semantics. The noise vectors z and z' are introduced as dropouts in both training and testing phases to avoid over-fitting of data. For our discriminator, we adopt the *Patch GAN* which aids in classifying real or synthetic instances by taking $(70*70)$ overlapping patches of images.

D. Evaluating the Closeness of Similarity between the Synthetic-Original Image

After the generation of fake images, we assess the similarity between the converted synthetic image

Authorized licensed use limited to: BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE. Downloaded on December 24,2024 at 00:52:26 UTC from IEEE Xplore. Restrictions apply.

Algorithm 1 Proposed network training

Input: Real image dataset of P domain, Reference model R , Generator model G_X and G_Y initialized with pre-trained model weights.

Output: D output with usual initialized weights

Assumptions: Number of iterations for training K and small-batch size B_s

for ($k = 1...K$) {

1. Select an arbitrary small batch of image samples x_{B_s} from training set P.

2. Make an augmented small-batch $\bar{X}_{B_s}$ with image samples $\bar{x}_{B_s}$ using distortion augmentation function.

3. Utilize $R(x_{B_s})$ to train D to predict $p_{B_s} \in$P. $Updatediscriminator$D$weightstominimizeeq.1$ $withη > 0$.
G(p$_{B_s}$) maps 2 functions: $G_X : P \rightarrow Q$ and $G_Y : Q \rightarrow P$.

4. Utilize $G_X(p_{B_s})$ and $G_Y(q_{B_s})$ to train corresponding D to predict $\bar{p}_{B_s} \in \bar{P}$ and $\bar{q}_{B_s} \in \bar{Q}$ respectively. Update weights of corresponding D to minimize eq.(1) with $\eta > 0$.

5.$Train$G$_X$ and G_Y to predict $G_X(p_{B_s})$ and $G_Y(q_{B_s})$ such that $D(G_X(p_{B_s}))$ and $D(G_X(q_{B_s}))$ predicts $p_{B_s} \in$P$and B_s \in \bar{Q}$. Update weights of G_X and G_Y to minimize GAN adversarial equation with $\eta > 0$.

}

and the original image. The extent of the similarity between the original and fake image can be measured by comparing the pixel values of the images and for this the correlation of the pixels is determined. For this purpose, we design *Algorithm 2* where the rendering of this image is done into high-dimensional pixel data. The pixel difference between the original and synthetic images is calculated by the Manhattan distance between the two images. The similarity of the two images is assessed by converting RGB value of every pixel to its gray values which are visualized as coordinates in the *n-dimensional* space. To model this process, we design an algorithm for evaluating pixel differences between two images for obtaining the similarity index of the original and fake images. Man_{dist}: Manhattan distance calculated between original and fake image. The Manhattan distance range is as follows: $(0, bits_{max}\sqrt{w_{orig} \times h_{orig}})$, where, $bits_{max}$: highest value of bits for every channel, w_{orig}: picture width, h_{orig}: picture height. The procedure *CalPVal()* takes an image and returns the RGB values associated with each pixel. It returns a list of n tuples where n is the no. of pixels in a picture. Each tuple is of size *3* and represents the R, G, and B values. The function *ConvRtoG()* converts the RGB values to grayscale values. *ManhattanDistance()* returns the normalized Manhattan distance between two images. The last procedure *Similar(Orig, Syn)* determines the similarity index of the images. *Algorithm 1* explains that a small batch of images is selected for introducing corruptions or distortions in the training set. The generator produces these variations and the discriminator predicts whether the instance belongs to the training domain or not by mapping the two domains shown in *step 4*. For testing the resiliency of SSD and Detectron different seasons are simulated such as $summer \rightarrow rainy$, and $summer \rightarrow night$. In addition to this, different artistic styles and data augmentations are considered from various viewpoints.

III. Data Generation

We take three approaches in comparison to the above-mentioned season transfers to mitigate the insubstantiality of our object detection models as compared to diverse synthetic falsifications. The three approaches are explained as follows: ***Approach 1 and Approach 2 (A1 and A2)***: We integrate stylized images with different style renderings namely: Vangogh which contains 400 paintings transforms an input sample in the Vangogh sample style and Ukiyoe contains 563 paintings. ***Approach 3 (A3)***: The AdaIN styles are integrated into the weather-simulated environments along with the augmented images. The augmented images satisfy the invariance in the unseen testing set and the adopted data augmentation techniques are as follows: *InvertTransformed* and *GrayScaleTransformed*. The robustness and exhaustive evaluations are shown in *Table I*.

IV. Model Training and Experimental Settings

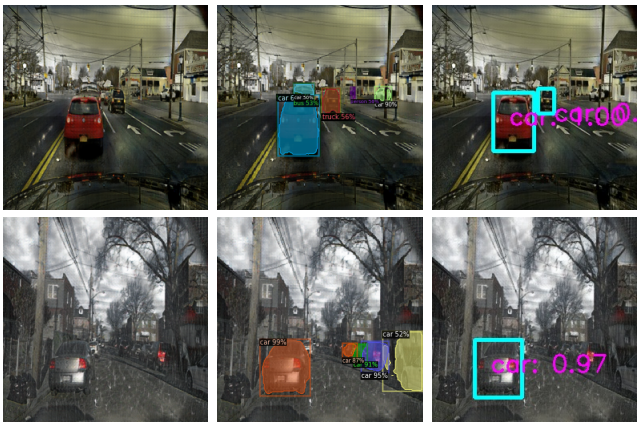

Figure 2: Object Detection in (a) summer to night (b) summer to rainy

Our proposed model is trained over new parameters and rendered to the different domain shifts which are assumed subtle to human observers. For example,

Authorized licensed use limited to: BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE. Downloaded on December 24,2024 at 00:52:26 UTC from IEEE Xplore. Restrictions apply.

Algorithm 2 Similarity between original and synthetic image

Input: Two Images *Orig* and *Syn*
Output: A Boolean value for similarity indicator between *Orig* and *Syn*

Procedure `Similar(`$Orig, Syn$`)`**:**
$Sim_{index} \leftarrow$ Similarity Index
Load $Orig$ Image Load Syn Image $w_{orig} \leftarrow$ width of $Orig$, $h_{orig} \leftarrow$ height of $Orig$, $bits_{max}$ $\leftarrow$ highest value of bits for every channel
$L1 \leftarrow$ CalPVal($Orig$)
$L2 \leftarrow$ CalPVal(Syn)
$gr1 \leftarrow$ ConvRtoG($L1$)
$gr2 \leftarrow$ ConvRtoG($L2$)
// calculate the distance between $Orig$ and Syn image using Manhattan metric. $N_{gr1} \leftarrow$ size of $gr1$ // $gr1$ and $gr2$ are of same size
$dist$ = 0
for ($i : 0;\ i < N_{gr1} - 1;\ i+ = 1$) {
  $dist \leftarrow dist + \sqrt{gr1_i - gr2_i}$
}

$$normalization \leftarrow \frac{d(a,b)}{bits_{max}\sqrt{w_{orig} * h_{orig}}}$$

// normalize Manhattan distance parameter
$Sim_{index} \leftarrow 1 - normalization$
if Sim_{index} *is above threshold* **then**
  return 1
end
else
  return 0
end

Procedure `CalPVal(`*Img* I`)`**:**
$N_p \leftarrow$ Img $I's$ pixels
$output \leftarrow$ null list of size N_p
for ($i : 0;\ i < N_p - 1;\ i+ = 1$) {
  $rval \leftarrow$ an empty tuple
  $rval$ stores RGB values of image pixels
  Append $rval$ to $output$
}
return $output$

Procedure `ConvRtoG(`*List* L_k`))`**:**
$N_{RtoG} \leftarrow L_k size$
$gList$,$emptylistofsizeN_{RtoG}$
for ($i : 0;\ i < N_{RtoG} - 1;\ i+ = 1$) {
  $gListL_k \leftarrow$ *0.3 * $L_{k,0}$ + 0.587 * $L_{k,1}$ + 0.114 * $L_{k,2}$*
  return gList
}

training the model under a single domain may yield insufficient results owing to the distribution shift in the pixel-level values of scenes. Further, the system settings for conducting these experiments are having specifications *Intel® Xeon® CPU @ 2.20GHz* with Core i7 9th Gen and a *Nvidia GeForce GTX 1080 Ti* GPU. Moreover, we have tuned the hyper-parameters such as *batch-size=2*, *n-epochs=120*etc.

Figure 3: Testing results of detection by methods Detectron and SSD on style transferred images. The stylized images chosen are of 2 different types: (a)Vangogh (b) Ukiyoe

Figure 4: Testing results of Detectron and SSD on augmented images with two (InvertTransform and Grayscale) augmented styles.

A. Training scenarios

We take four setups for our object detection models to perform in different domains which are as follows: ***Setup 1 (S1):*** SSD and Detectron are fed with SYNTHIA dataset images. ***Setup 2 (S2)***: In this scenario, we have aggregated image samples gathered from the Internet source to the PASCAL VOC dataset. ***Setup 3 (S3):*** In the third scenario, we observe the augmented images. ***Setup 4 (S4):*** We incorporate two weather modellings i.e. $summer \rightarrow rainy$ and $summer \rightarrow night$.

Authorized licensed use limited to: BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE. Downloaded on December 24,2024 at 00:52:26 UTC from IEEE Xplore. Restrictions apply.

Table I: Varied sets of training approaches and setups for robustness and exhaustive investigations of object detectors

Types of Datasets		**Robustness Evaluation**			**Exhaustive Evaluation**			
		A1	**A2**	**A3**	**S1**	**S2**	**S3**	**S4**
BDD		✓	✓	✓	✓	✓	✓	✓
Basic Augmentation	**Gray Scale Transformed**	✓	✓	✓	✓	✓	✓	✓
	Invert Transformed	✓	✓	✓	✓	✓	✓	✓
Summer->rainy(s2r)		✓	✓	✓			✓	✓
Summer->night(s2n)		✓	✓	✓			✓	✓
Ukiyoe style				✓				
Vangogh style		✓						

V. Experimental Results

The discussed three approaches and four setups are assessed on their efficiency and how well they can generalize well to the diverse variations provided in our database of images. *Detection Results in Combined dataset*: The combination of PASCAL VOC+ augmented+ stylized images + proposed model generated synthetic images using SYNTHIA are used for testing the images. *Figure 2* shows two types of weather transformations modeled in these images. SSD and Detectron show good performance on these new synthetic variations. We also compute the average precision (AP) of these object detectors to authenticate their performance using two different threshold levels (50,75) in *Table III*. *Stylized images*: The three approaches considered for our experiments comes under AdaIN fast-style transfer method. *Figure 3* displays the testing results of the object detectors when tested on the Vangogh and Ukiyoe styles. *Testing Results of Augmented Images*: To keep the fairness in the detection process, we did not include the augmented images in the training set.

VI. Robustness Evaluation

After training the object detectors with proposed model-generated images and with other domain shifts we evaluate the model performance using the mean performance degradation for the corrupted testing set (mPD) at a threshold of 50 IoU. Moreover, the relative performance degradation under corruption (rPD) is also taken into account shown in *equation 1* and *equation 2* as follows:

$$mPD = \frac{1}{L_s} \sum_{s \in S} AP^s_{50} \qquad (1)$$

$$rPD = \frac{rPD}{AP^{ns}_{50}} \qquad (2)$$

where, L_s is the total number of severity levels, AP^s_{50} signifies mean average precision under distortion with s level of severity, AP^{ns}_{50} shows mAP of our model on real data having zero variations. The results of discussed assessment approaches for the object detector's resiliency are shown in *Table II*. *Approach 1 and Approach 2 (A1 and A2)*: *Table III* validates the performance of the object detectors on two style renderings and two augmentations by showing good values signifying less vulnerability of SSD and Detectron. It is because of the less inclination to the texture bias which is generated by the styled variations and forces the detector models to learn the objects based on shapes. *Approach 3 (A3)*: Unlike, *A1 and A2* the AdaIN renderings in the weather simulated images in *Figure 3* have shown better results in proving the robustness of the detector models. The augmented images in *Figure 4* also displays better performance which is evidenced in the *Tables III*.

A. Exhaustive Evaluation

The results under four different setups are as follows: *Setup 1 (S1)*: Detectron outperforms SSD at the cost of computational speed. *Setup 2 (S2)*: Internet images and PASCAL VOC images are combined together for training our models. *Setup 3 (S3)*: Testing the models on the augmented images enhance the average precision of both models in all cases. *Setup 4 (S4)*: *Table III* validates that introducing weather transformations has improved the results of the models. We also utilize the loss function given by Wasserstein GAN (WGAN) as opposed to the conventional loss format advocated by original GAN. *Figure 5 (a)* and *Figure 5 (b)* signify Generator's loss and discriminator loss which displays an erratic distribution. The cycle consistency in *Figure 5 (c)* helps learn different domains while retaining the common content.

Table III: Calculated results of SSD (model 1) and Detectron (model 2) in the robustness evaluation for two severity levels

Models	**S=1**	**S=2**	AP^{ns}_{50}	**mPD**	**rPD**
SSD	68	63	77	65.5	85.06
Detectron	72	66	82	69	84.14

Authorized licensed use limited to: BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE. Downloaded on December 24,2024 at 00:52:26 UTC from IEEE Xplore. Restrictions apply.

Table II: Mean average precision results of SSD and Detectron model

Scenarios	**SSD**		**Detectron**	
	AP_{50}	**Mean AP (50,75)**	AP_{50}	**Mean AP(50,75)**
PascalVOC	77	64.5	82	67.5
Augmented	68	54	72	57.5
Adain Style Transfer	63	47.5	66	51
s-w Cycle GAN Translated	70	57	77	62.5
s-n Cycle GAN Translated	71	57.5	78	63.5
s-r Cycle GAN Translated	70	56.5	76	61.5

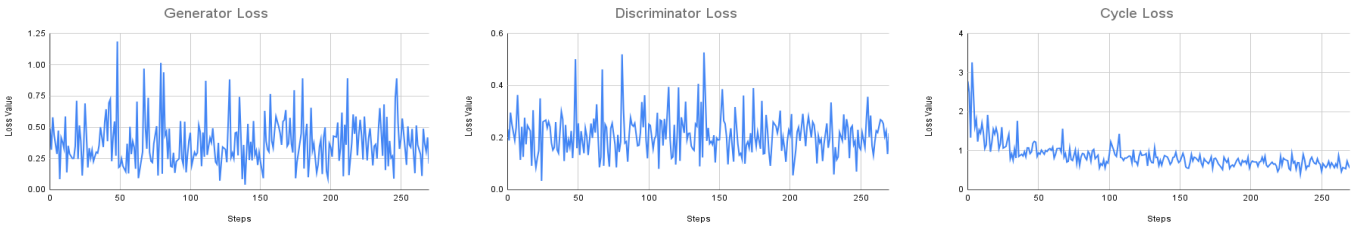

Figure 5: Loss functions of (a) Generator (b) Discriminator (c) Cycle consistency

VII. Conclusion

In this paper, we have proposed a novel framework which learns the inter-domain adaptations in the absence of labeled image pairs to provide a diversified dataset for testing the performance of the state-of-the-art object detection models. The testing results of the weather-translated images generated by the proposed framework along with different stylized and augmented images have shown improved detection capability of SSD and Detectron giving rPD values 85.06% and 85.14% respectively, thereby showing a gain of approximately 12% over the standard dataset. Furthermore, the exhaustive evaluation has shown comparable mAP amongst all the variations.

Acknowledgment

This work is supported by CHANAKYA Fellowships of IITI DRISHTI CPS Foundation under the National Mission on Interdisciplinary Cyber Physical System (NM-ICPS) of Department of Science and Technology, Government of India.

References

[1] Amir Khosravian, Abdollah Amirkhani, Hossein Kashiani, and Masoud Masih-Tehrani. Generalizing state-of-the-art object detectors for autonomous vehicles in unseen environments. *Expert Systems with Applications*, 183:115417, 2021.

[2] Di Feng, Christian Haase-Schütz, Lars Rosenbaum, Heinz Hertlein, Claudius Glaeser, Fabian Timm, Werner Wiesbeck, and Klaus Dietmayer. Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *IEEE Transactions on Intelligent Transportation Systems*, 22(3):1341–1360, 2020.

[3] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3234–3243, 2016.

[4] Yuhu Shan, Wen Feng Lu, and Chee Meng Chew. Pixel and feature level based domain adaptation for object detection in autonomous driving. *Neurocomputing*, 367:31–38, 2019.

[5] Tejas S Borkar and Lina J Karam. Deepcorrect: Correcting dnn models against image distortions. *IEEE Transactions on Image Processing*, 28(12):6022–6034, 2019.

[6] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: high quality object detection and instance segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 43(5):1483–1498, 2019.

[7] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[8] Renwan Bi, Jinbo Xiong, Youliang Tian, Qi Li, and Kim-Kwang Raymond Choo. Achieving lightweight and privacy-preserving object detection for connected autonomous vehicles. *IEEE Internet of Things Journal*, 2022.

[9] Che-Tsung Lin, Sheng-Wei Huang, Yen-Yi Wu, and Shang-Hong Lai. Gan-based day-to-night image style transfer for nighttime vehicle detection. *IEEE Transactions on Intelligent Transportation Systems*, 22(2):951–963, 2020.

[10] Prateek Chhikara, Rajkumar Tekchandani, Neeraj Kumar, Vinay Chamola, and Mohsen Guizani. Dcnn-ga: A deep neural net architecture for navigation of uav in indoor environment. *IEEE Internet of Things Journal*, 8(6):4448–4460, 2020.

[11] Aryan Mehra, Murari Mandal, Pratik Narang, and Vinay Chamola. Reviewnet: A fast and resource optimized network for enabling safe autonomous driving in hazy weather conditions. *IEEE Transactions on Intelligent Transportation Systems*, 22(7):4256–4266, 2020.

[12] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

Authorized licensed use limited to: BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE. Downloaded on December 24,2024 at 00:52:26 UTC from IEEE Xplore. Restrictions apply.