Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 7479473, 11 pages
https://doi.org/10.1155/2021/7479473

Research Article

Generating Bird's Eye View from Egocentric RGB Videos

Vanita Jain,[1] Qiming Wu,[2] Shivam Grover,[1] Kshitij Sidana,[1] Gopal Chaudhary,[1] San Hlaing Myint,[3] and Qiaozhi Hua[4]

[1]*Bharati Vidyapeeth's College of Engineering, New Delhi, India*
[2]*China Mobile (Hangzhou) Information Technologies Co., Ltd., Hangzhou, China*
[3]*Global Information and Telecommunication Institute, Waseda University, Tokyo, Japan*
[4]*Computer School, Hubei University of Arts and Science, Xiangyang, China*

Correspondence should be addressed to Qiaozhi Hua; 11722@hbuas.edu.cn

Received 3 August 2021; Revised 27 September 2021; Accepted 16 October 2021; Published 8 November 2021

Academic Editor: Jerry Chun-Wei Lin

In this paper, we present a method for generating bird's eye video from egocentric RGB videos. Working with egocentric views is tricky since such the view is highly warped and prone to occlusions. On the other hand, a bird's eye view has a consistent scaling in at least the two dimensions it shows. Moreover, most of the state-of-the-art systems for tasks such as path prediction are built for bird's eye views of the subjects. We present a deep learning-based approach that transfers the egocentric RGB images captured from a dashcam of a car to bird's eye view. This is a task of view translation, and we perform two experiments. The first one uses an image-to-image translation method, and the other uses a video-to-video translation. We compare the results of our work with homographic transformation, and our SSIM values are better by a margin of 77% and 14.4%, and the RMSE errors are lower by 40% and 14.6% for image-to-image translation and video-to-video translation, respectively. We also visually show the efficacy and limitations of each method with helpful insights for future research. Compared to previous works that use homography and LIDAR for 3D point clouds, our work is more generalizable and does not require any expensive equipment.

1. Introduction

Egocentric videos, commonly referred to as first-person videos, are captured from the POV of a subject (in our case from the POV of an autonomous vehicle). Egocentric videos are easy to capture and hence are accessible in real-time to the vehicle. However, they are deviously hard to for a computer to comprehend and work with. This is because egocentric videos are prone to occlusions, and there is a significant warping effect due to perspective which causes the objects closer to the camera to look inflated. Another drawback of the egocentric view is the nonlinear nature of objects in motion.

On the other hand, top-down views such as the views from a surveillance camera or drones show a more holistic and consistently scaled view of the environment, which makes them rich in data and easy to work with (see Figure 1). Previous work done in fields such as trajectory prediction is mainly focused on CCTV footage. State-of-the-art methods work irrespective of view but perform much better at top-down views of 45°or greater. Not only that, increasing the angle from 0° (eye level) to 90° (top-down) eliminates most of the occlusions and improves the visibility. With advancements in self-driving autonomous vehicle technology, it becomes important that we devise a way to overcome the shortcomings of egocentric perspective and make their accessibility useful [1–3].

In this paper, we present an approach for generating a bird's eye view of the environment from egocentric images. Unlike previous works [1, 4, 5] that use homography and/or perspective transform for estimating the coordinates of objects in a bird's eye view, we majorly aim to reconstruct the whole visible scene including the objects of interests (such as cars and pedestrians) and all other objects (such as buildings, trees, and crosswalks) that may affect the future behavior of the objects of interests. Our work is aimed at

Figure 1: Example pairs of egocentric views (left row) and their corresponding bird's eye view (right row). The egocentric views are highly warped due to perspective and a major part of the environment is out of the field of camera's view. Bird's eye view shows a holistic view of the environment, and the scaling is consistent.

maintaining geometrical, spatial, and temporal consistency during the view translation. To the best of our knowledge, this has been an unexplored domain [6].

We identify this as a problem of view translation, and it can be solved through image-to-image or video-to-video translation, each having its own perks and shortcomings. We show experiments with both approaches and give directions for future research. We use an adversarial approach for the deep learning model takes as input an egocentric image and learns to generate its corresponding bird's eye view. Our work opens new avenues for progress in self-reliant and smarter autonomous systems [7, 8]. This also enables the development of smarter connected vehicles. Having egocentric views from multiple nearby vehicles, a much more accurate prediction of bird's eye view will be possible which is an area of interest for future research [9].

With the advent of Industry 5.0, interconnection of not only devices but also vehicles will be possible. Vehicles in proximity can collaboratively develop the novel viewpoint and fill in blind spot caused by occlusions [10]. Our work acts as a stepping stone towards making this possible [11].

2. Related Work

2.1. Classical Approach for View Translation. Perspective transformation is a classical approach to compensate for the camera angle. Using homography [4, 12–18], a plane is resolved, and the transformation is applied to correct the perspective. Since this approach relies on a mathematical approach to the problem, the resulting image can appear to be distorted and out of proportion.

2.2. 3D Point Cloud for View Translation. With the availability of technologies such as Lidar that readily give the 3D point clouds for the scene, obtaining a bird's eye representation for various applications [19–23] is relatively simple as compared to using RGB image as input. The LIDAR gives a readymade 3D point cloud of the environment which after some processing can be transformed into a 2D view from any specific angle. However, such sensors are expensive, and not all vehicles are equipped with them. Dashcams and cameras installed on mobile phones are generally incapable of inferring the 3D information and only provide RGB images. Our method uses a single RGB image as input, thus eliminating the use of any expensive equipment.

2.3. Learning-Based Approach. Learning-based approaches have been gaining popularity as they provide promising results in similar applications. This majorly includes those approaches in which we train our system to learn from a predisposed set of data. Convolutional neural networks have impacted the domain of image analysis greatly, and consequently, there have been works that use CNN along with other traditional methods such as homography to have a more dynamic approach towards generating bird's eye view from a single image. [18] uses a CNN to predict 4 parameters of the homography matrix which is used to transform the image into its bird's eye representation further. However, their model is majorly for images that already have some vertical leverage (for example from CCTV cameras) and would not be able to work on egocentric images such as those coming from a dashcam of a car, where the views are highly skewed with little scope for homography to work. In our work, we show an end-to-end approach for translating nonvertical egocentric images into their corresponding bird's eye views using a completely learning approach.

3. Methods

3.1. Dataset. We needed a dataset that has egocentric images (from a car's point of view) along with their corresponding bird's eye views. This poses three major constraints for bird's

eye views (see Figure 2). (1) The pixel position of the subject car in all of bird's eye frames should be the same, in a way that the rest of the environment appears to be moving and the subject car appears to be stationary. (2) The camera angle in bird's eye view should also be such that a vertical line through the centre of the image should pass through vehicle's body perpendicularly. (3) The distance of the top-down camera from the car should also remain constant. A dataset satisfying all three of these requirements will allow for a consistent representation and avoid any discrepancy regarding the alignment and position of the camera during the image generation process.

Such a dataset is extremely hard to curate in the real world. Capturing the egocentric feed is easy and can be achieved by simply placing a fixed camera inside the car or on car's body. However, capturing bird's eye view is nearly impossible, especially with the constraints mentioned above. A plausible approach may be using a drone camera that hovers over the car. But keeping it stationary relative to the car is practically impossible.

So, we decided to make use of synthetic data for training purposes. Advances in graphics technology offer us hyperrealistic animation and games that we can make use of as an alternative for real-world data. One such game is Grand Theft Auto V (GTAV) in which the visuals of the environment and the behaviour of the cars and pedestrians mimic that of the real world. We make use of the SVA dataset released by [24] in which the camera changes between egocentric and bird's eye view at alternate time steps, which gives a highly accurate bird's eye representation for each egocentric frame. The camera also follows the constraints we mentioned above. Two sample sequences from the dataset can be seen in Figure 3. While the dataset released by [24] also contains bounding boxes, yaw, and other relevant information for nearby cars, we do not include that into our training process and leave that to future work.

3.2. View Translation. Before building a system, it is necessary to understand the data from the egocentric images that we would like to retain in bird's eye views. Taking the case of the view from a dashcam of a car, we not only want the objects of interest such as other cars and pedestrians to appear in bird's eye view but also the other aspects of the environment that may affect our or other cars and pedestrians' behaviours. For this, simply projecting the coordinates of the objects of interest in a top-down view is not enough. To this end, we treat this as a problem of view translation where we try to retain as much information as possible and describe how we achieve it below.

Image to image translation [25–35] is one such approach that generates images in one domain using images from another domain. This approach is best suited for isolated frames or images as it lacks temporal consistency. Video to video translation [36] is similar to image-to-image translation but improves upon temporal consistency [37]. We talk about how we made use of these in our work and how well they perform compared to each other.

The major task at this point is to generate a bird's eye view y given an egocentric input x. Generative adversarial networks (GANs) [38] have performed remarkably well in the deep learning-based generative area of study. The architecture of a GAN consists of two parts: a generator G and a discriminator D. The generator is supposed to generate unseen but realistic data that falls in a similar domain as the training dataset, and the job of the discriminator is to classify a generated data point as realistic or fake. G and D are both trained together in a two-player min-max situation, where we try to establish a Nash equilibrium. But simple GANs are only effective in generative image synthesis applications if we need to generate new examples of images. We basically have no control over the data being generated [39]. To be able to control the outputs and to make use of additional information, such as class labels, or in our case, an input image of egocentric domain x that we want to be translated into an image of bird's eye domain y, we use an extension of GANs called conditional GANs [40, 41].

In conditional generative adversarial networks, the generator G learns to generate fake samples with a conditioned data point of domain x instead of unknown noise distribution as in simple GANs. The final objective of a conditional GAN looks like the following:

$$\mathscr{L}_{\text{cGAN}}(G, D) = \mathbb{E}_{x,y}[\log D(x, y)] + \mathbb{E}_{x,z}[\log (1 - D(x, y))]. \tag{1}$$

In the task of image-to-image translation, the condition input is an image of domain x, and the generator outputs its corresponding image in the search space of domain y. There has been quite some progress in the field of image-to-image translation when combined with conditional GANs. Conditional GANs for image-to-image translation has been used to achieve tasks like colourization of black and white images by Zhang et al. [42], future frame prediction [43, 44], and image prediction from normal maps [45, 46]. We build on the work by Isola et al. [26] which consists of a general image to image translation network. They also incorporate a convolutional PatchGAN classifier for the discriminator which allows the structure to penalize at the scale of image patches. So, instead of trying to check whether the image as a whole is real or not, the PatchGAN checks whether each $N \times N$ patch in the image fed to the discriminator is real or not [47]. Then, the predictions by the discriminator for all patches are averaged and given out as the final output.

Along with the cGAN loss in Equation (1), they also use a traditional L1 loss. This forces the generator to generate images near the ground truth output in an L1 sense while also trying to fool the discriminator into believing the generated images are real.

$$\mathscr{L}_{L1}(G) = \mathbb{E}_{x,y,z}[\|y - G(x, y)L1\|. \tag{2}$$

This results in their final objective function as

$$G^* = \arg \min_G \max_D \mathscr{L}_{\text{cGAN}}(G, D) + \lambda \mathscr{L}_{L1}(G). \tag{3}$$

Apart from the PatchGAN, their generator network uses

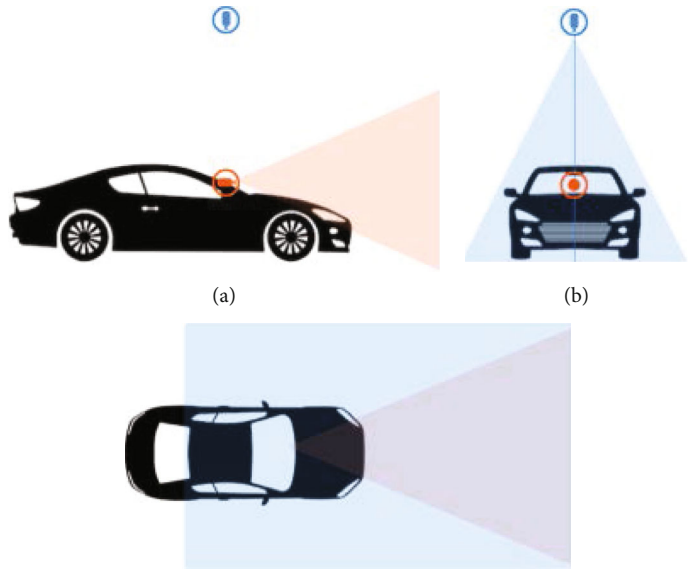

Figure 2: An overview of the camera placements for collecting the dataset. The camera with orange field of view captures the egocentric view, and the camera with blue field of view captures bird's eye view. The position (constant) and a range of the camera views (not to scale) are demonstrated as well.

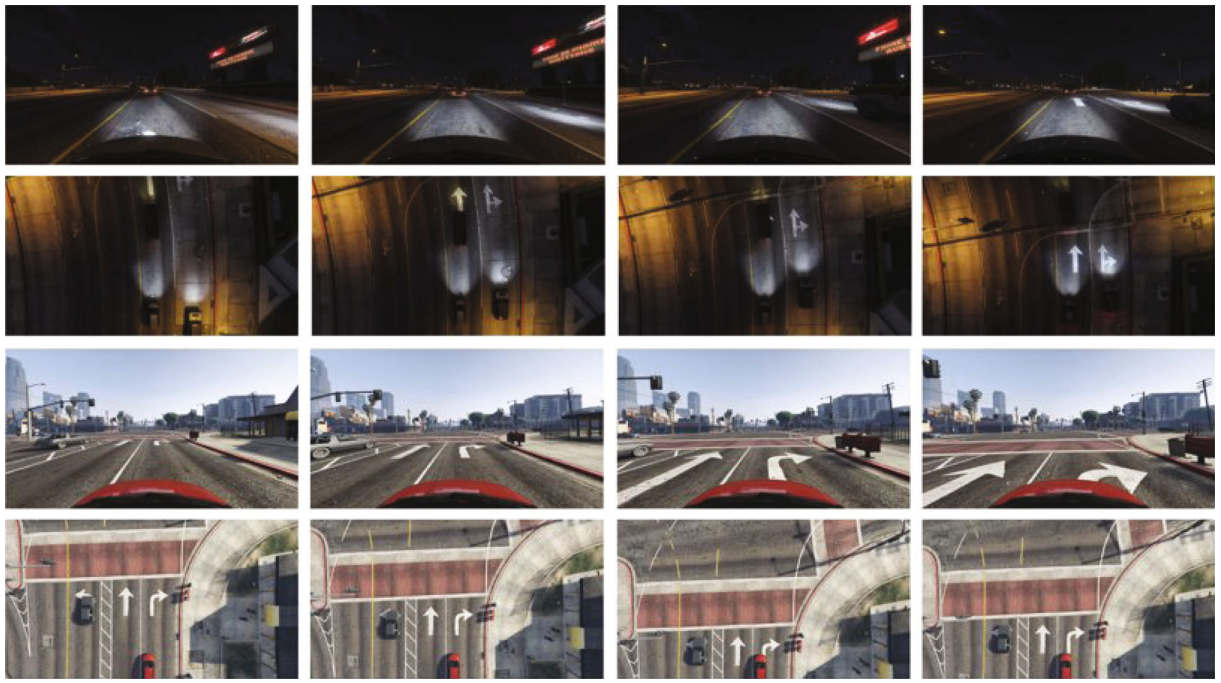

Figure 3: Sample images from the dataset we used. Each pair of rows are a different sequence, and the top row in each sequence shows the egocentric views, and the bottom shows bird's eye views.

a U-Net [48] style architecture which allows them to establish a better relationship between the input and output images that have the same low-level structure such as in image colourization and simulation to reality. In our case, this feature is not as useful since our input images and output images are considerably different, and this does not prove to be a disadvantage either as the network without the U-Net architecture gave similar results to the original Pix2Pix network. We also show the quantitative comparison of both in the next section.

For each step in training the network, we randomly pick an egocentric image from the sequences and give it as the input along with its corresponding bird's eye view as the ground truth label. An overview of the training process can

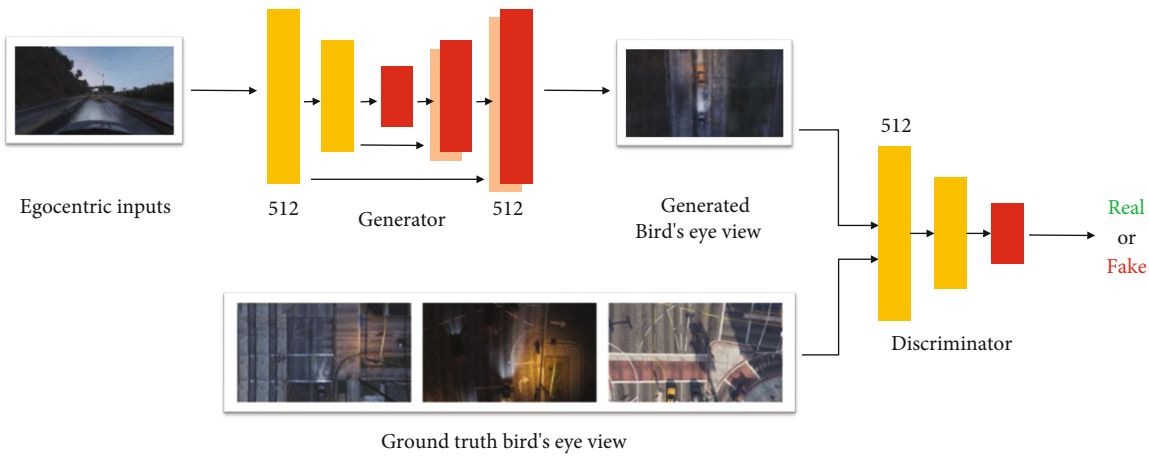

Figure 4: Training pipeline for image-to-image translation. The egocentric image is sent into the generator, and the generator outputs a predicted bird's eye view, which is compared to the ground truth view (not shown). To make the results look realistic, a discriminator is also trained simultaneously that predicts whether the generated image is real or not.

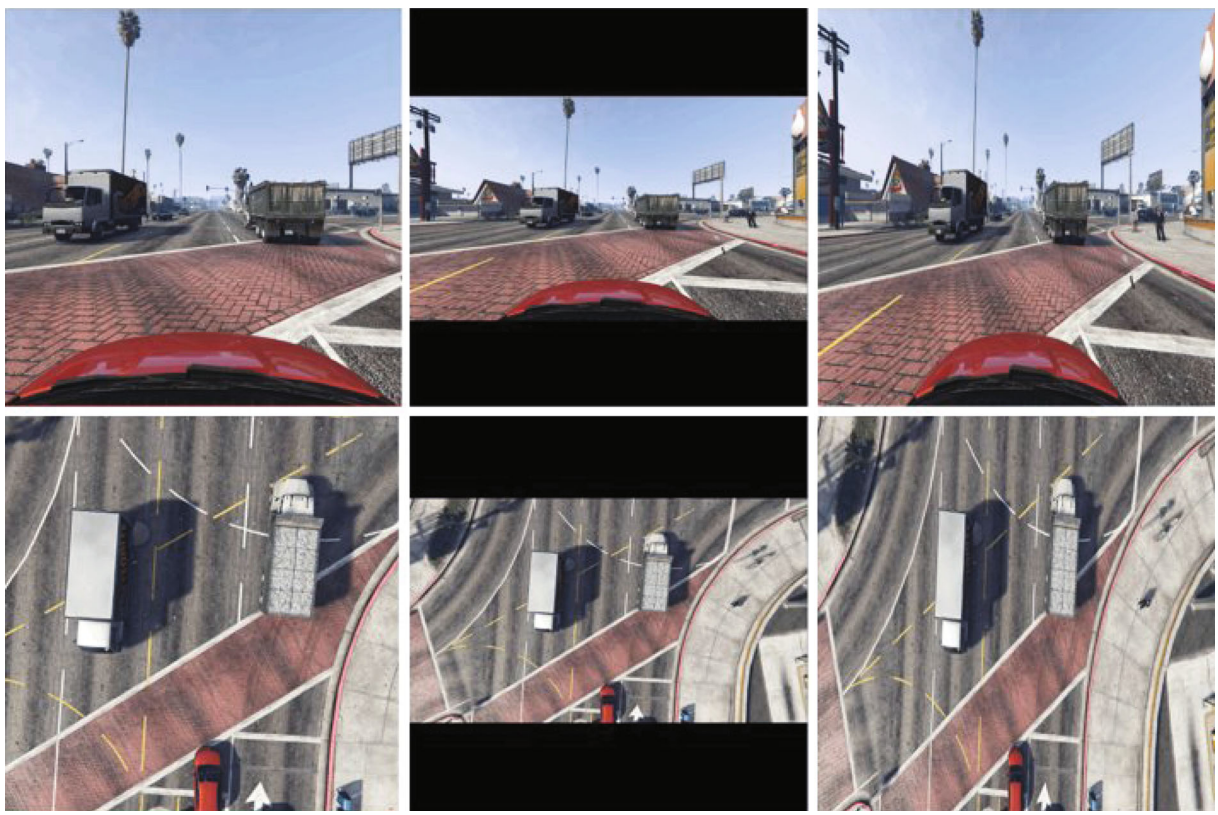

Figure 5: Each column represents three different methods for providing the egocentric (top) and bird's eye view (bottom) images to the image translation model.

be seen in Figure 4 The images were originally of the aspect ratio 16 : 9, whereas the network takes as input images with a 1 : 1 aspect ratio (or square images). To solve this, we could do three things as shown in Figure 5. (1) Centre crop the image as a square. This however leaves out the peripheral vision which is very important especially for autonomous vehicles since it is important to keep track of the vehicles that are trying to overtake you. (2) Add padding to the top and bottom of the original image so that it turns into a square. Unlike centre cropping, this does not leave out any information present in the original image. However, the issue with this was that almost a quarter of the image space was being wasted on padding. (3) Resize the image into a square. In this approach, no information is being left out, and the space is being utilised efficiently. The only issue with this method is that the internal aspect ratio of the original image is ruined which makes the image look squished. However, this does not seem to affect the learning of the model

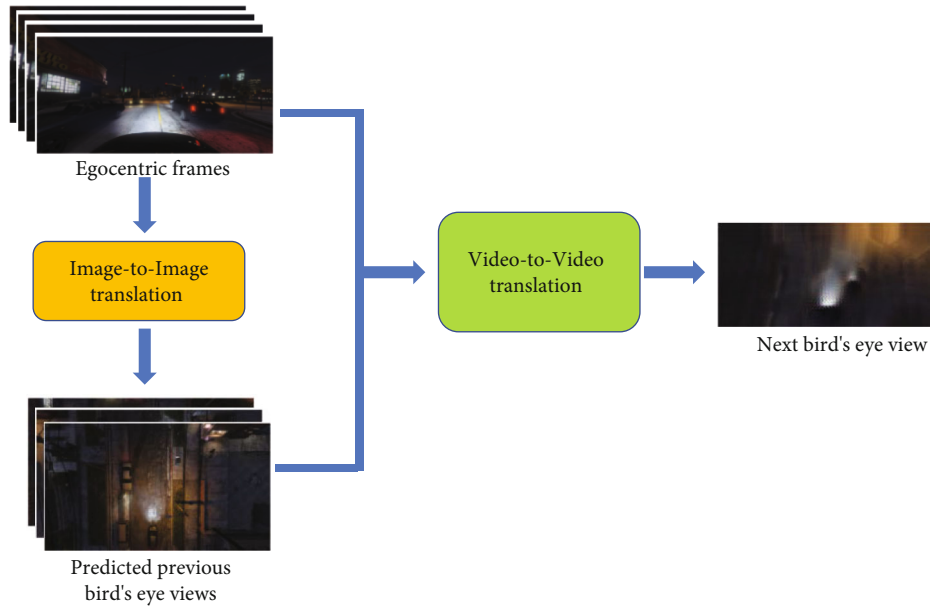

Figure 6: Training pipeline for video-to-video translation. For the first three egocentric frames, we use the image-to-image translation module to generate rough predictions of bird's eye view. All these along with the fourth egocentric views are input to the model, and the model generates bird's eye view for the fourth frame. Then, for the fifth frame, we also send the previously generated output as the label for the fourth frame, and this goes on until all frames have been processed.

negatively and seemed to be the most effective out of all three methods. Further efforts on evaluating the three different methods can be seen in the experiments section.

Since the application of our work is primarily in a video-based task, we also decided to use a temporally consistent model [36] for training. In this approach, the model requires us to send a sequence of frames as input instead of a single frame as we saw in image-to-image translation [26]. The model works in a coarse-to-fine way, i.e., first, a low-resolution model takes as input an image and along with a sequence of previous output images. For the very first image, we use the image-to-image translation model for generating the previous output images. Then, the generator outputs the next frame (Figure 6). Then, higher resolution generator is stacked on top of this generator which is used to increase the resolution of the generated frame. Once the model starts to predict the next frame in the sequence, we then use its predicted frame for subsequent inputs (this also deteriorates the input quality for the next frame, which might cause a significant cascading effect and the quality of the predictions decrease continuously) (Figure 7). We use images of sizes 1024×512 which require us to use two generators. The first one outputs images of size 512×256, and the second one gives us the final output.

4. Results

In this section, we will show and evaluate the results of our view translation pipeline.

4.1. Image-to-Image Translation. For image-to-image translation, the first experiment that we conducted was to establish the best method to crop and resize the images before feeding them into the model as ground truth. We checked three different methods of arranging them as seen in Figure 5. We trained a model three times on the same dataset and each time preprocessing the images differently. We did a qualitative and quantitative analysis for establishing which method is the best. For qualitative analysis, we did a user study with 5 human subjects and asked them to rate the generated images from each method on a scale of 1 to 10 on three factors: image quality, amount of crucial information retained, and the amount of details in the image (Table 1).

Note that for the third method, we resized the images into a square and sent that into the model. The generated image was a square as well. However, since it looked squished, we inverted the resize factor of the generated image back to their original aspect ratio so that they look natural to the users.

To quantitatively evaluate the different methods, we checked the mean structural similarity index and the root mean square error between the output images and their corresponding ground truth images on a test dataset containing 20 images. We show the average values in Table 2 where method I correspond to Figure 5 (first column), method II corresponds to Figure 5 (second column), and method III corresponds to Figure 5 (third column). After testing, it seemed intuitive to use method III for the final training.

Next, we show the results of the final image-to-image translation model on unseen input images in Figure 8. On comparing the generated results with the ground truth, we get the average SSIM value as 0.72 and RMSE value as 30.56. We also tested the model with the U-net with skip connections, and we got nearly the same results with an average SSIM value of 0.712 and RMSE value of 28.25. To quantitatively evaluate the details, present in the generated

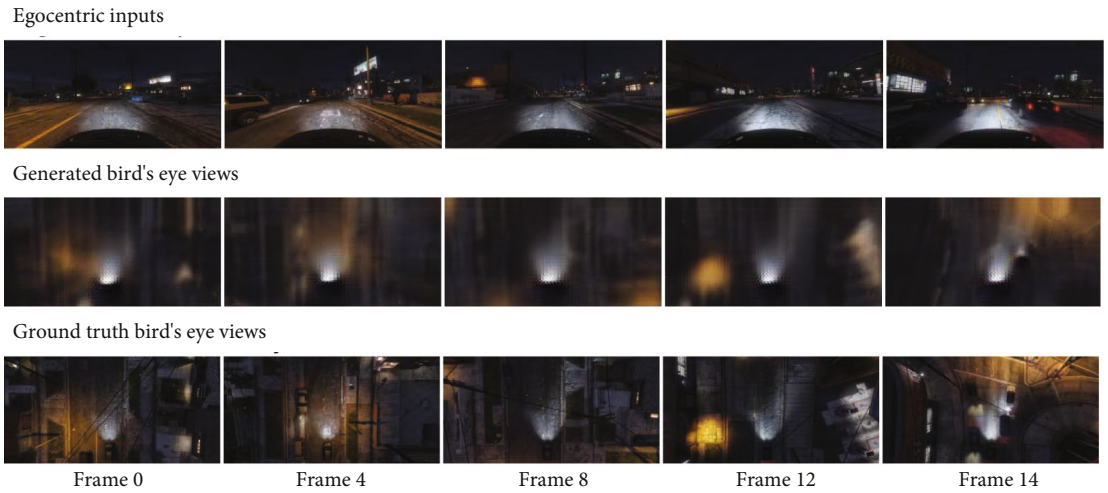

Figure 7: Results of the video-to-video translation experiments on a test sequence.

Table 1: A user study on different types of generated images from differences in preprocessing.

Factor	I	II	III
Image quality	8.5	6	8.5
Crucial information present	6.5	8	8.5
Details persevered	8.5	5	8

Table 2: Quantitative analysis of different types of generated images from differences in preprocessing.

	I	II	III
RMSE	30.1	35.4	32.19
SSIM	0.62	0.51	0.65

images, we further perform edge detection using a Canny edge detector on multiple predicted images and their corresponding ground truth images.

On comparing the ground truth edges with the edges in the generated images, we get an average SSIM score of 0.761 and an average RMSE score of 70.54. With the skip connections, we got an average SSIM score of 0.728 and an average RMSE score of 68.25. In Figure 9, we show three good results (retained most of the useful details, such as shapes of cars and crosswalks) bounded with a green and three failure cases (did not retain much useful details) bounded with a red box. The model is even able to understand subtle details such as the headlights being on in the vehicles. On a visual observation of the generated images, the results seem blurry and do not quite capture the environment exactly as in the ground truth images. This is a limitation of the type of model we selected, and we talk more about this in the discussion section and also mention the research areas that might help in tackling this issue. In Figure 10, we also compare our results with the results obtained from homographic transformations. We compare the nonblank parts of the image with the corresponding parts in the ground truth image and get an average SSIM of 0.41 and an average RMSE of 47.0. Compared to the homographic results, our image-to-image translation results are better by a margin of 77% for SSIM and 40% for RMSE. Visually, the homographic results look very distorted, and the objects cannot be reliably detected.

Figure 8: Sample results of the image-to-image translation method. Bounding boxes are given in second and third columns to show the actual positions of vehicles. A green bounding box signifies a successful reconstruction including the position of the reconstructed vehicle, whereas a red bounding box signifies an unsuccessful or missing reconstruction.

We finally conducted experiments for video-to-video translation. In Figure 7, we show the results for a test sequence of 14 frames. The model is able to reconstruct bird's eye view and successfully captures details such as

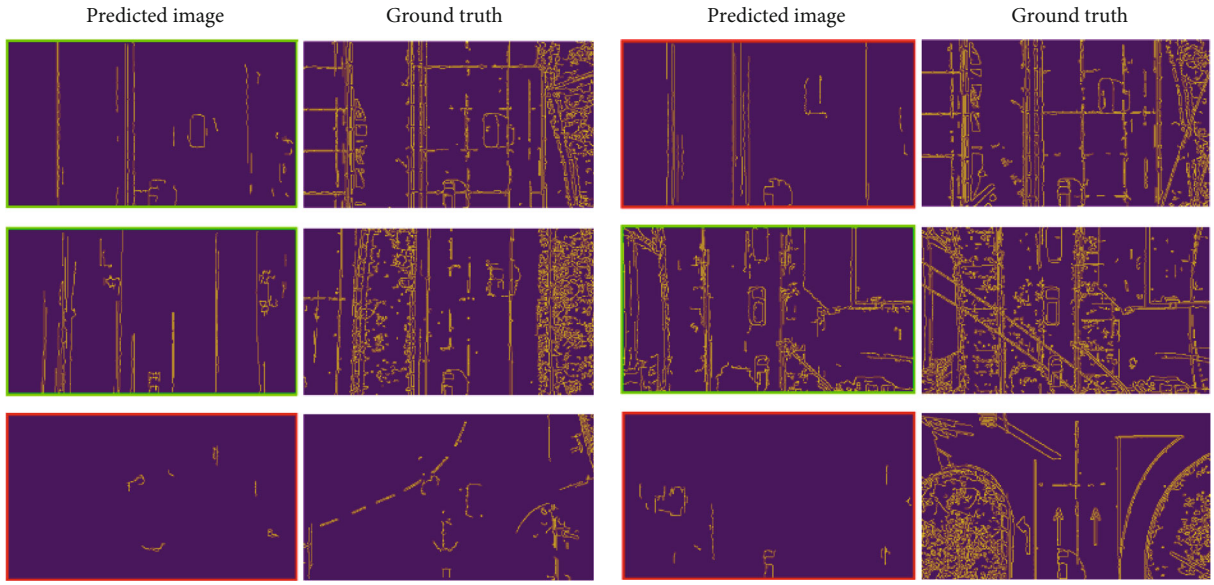

Figure 9: Comparison of detected edges on predicted and ground truth bird's eye view images. The generated images outlined with green retain the useful details such as shapes of cars, roads, and buildings, whereas the generated images outlined with red fail to retain useful details.

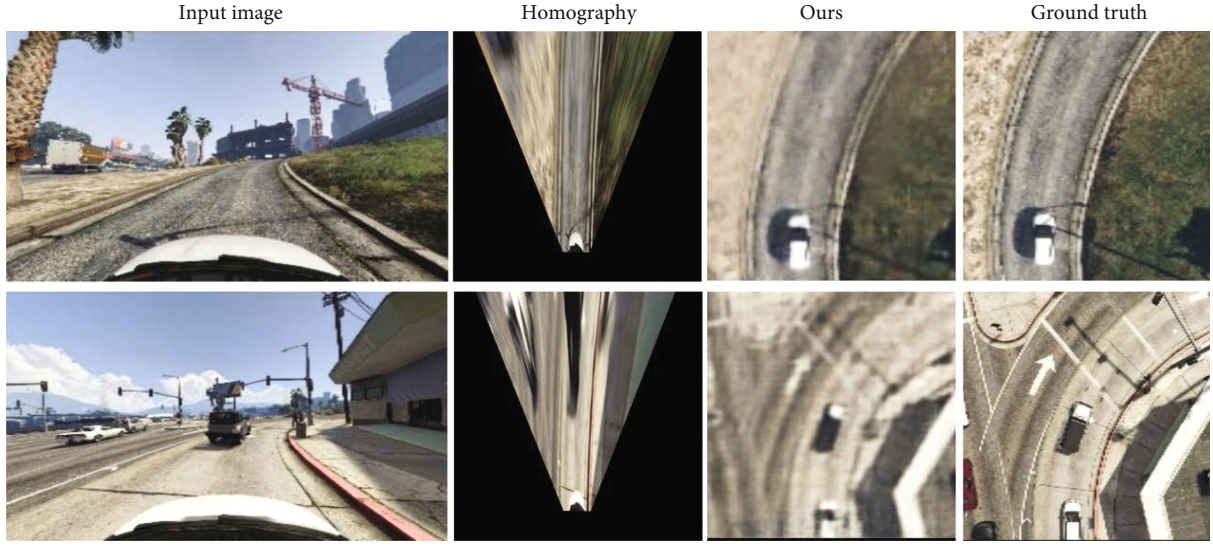

Figure 10: Comparison of our method with homography.

nearby cars, headlights, and ambient lights in a temporally consistent manner. On the negative side, the results are blurry. We talk about why this is so in the discussions section below and also mention the possible solutions. Initial results are better than the future frames, and the details start to deteriorate as more frames are predicted by the model. This happens because, for each consequent step, the model takes as input the previously generated frame, which propagates the errors forward deteriorating the quality of each consequent image. To evaluate the results quantitatively, we compared the generated bird's eye views and their corresponding ground truths and got the calculated RMSE value as 40.25 and the SSIM as 0.47. Compared to the homographic results, our video-to-video translation results are better by a margin of 14.4% for SSIM and 14.6% for RMSE.

We also show the comparison between the two methods in Figure 11. We ran both models on the same set of 6 frames of multiple sequences. In Figure 11(a), we show the abilities of the model to generate images that are similar to the actual ground truth. For this, we simply calculate the SSIM values of each generated image and its corresponding ground-truth bird's eye view. The SSIM values for image-to-image translation do not follow any trend; however, the values for video-to-video translation degrade as more frames are generated. This is due to the cascading effect on errors in each generated frame being propagated forward. In

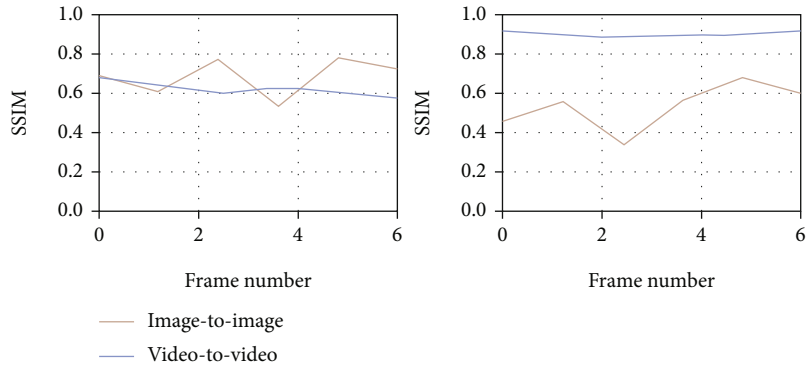

Figure 11: Comparison of the results from image-to-image and video-to-video translation methods. (a) The SSIM values of each generated frame with its corresponding ground-truth frame. The SSIM values in (a) for the image-to-image method do not seem to follow any trend, whereas for the video-to-video translation method, the quality of the image seems to degrade a little as more frames are generated. (b) The SSIM values of each generated frame with its previous generated frame. In (b), the consecutive frames from image-to-image translation show little similarity, whereas the consecutive frames from video-to-video translation show high similarity and hence consistency.

Figure 11(b), we compare the consistency and similarity in the consecutive frames generated from both methods. For this, we find the SSIM between a generated frame and the frame generated before it. It should be noted that even in the most ideal case, the value will never be 1 as the temporal change in the egocentric images will always incur a change in bird's eye view. However, a high value still shows that there is a good level of consistency in the consecutive frames. Video-to-video translation shows high levels of consistency, whereas image-to-image translation gives low SSIM values.

5. Discussions and Future Works

Our work shows the possibility of using RGB egocentric images for inferring bird's eye view around the subject vehicle. The failure results of work also provide key insights and directions that may benefit future researchers. Architectures such as [26, 36] work better for translations that have some level of geometric alignment, for example, horse-to-zebra or oranges-to-apples, where the input image and the output image are geometrically and structurally very similar, with differences only in the appearances and textures. However, in the task that we aimed to solve, there is a high level of geometric deformation in the input and output images. Egocentric images are completely different from top-down images, and even though this difference is consistent in all such images, models such as [26, 36] are not well-equipped for this. In order to solve the issue of geometric deformation in such images, future works may look at deformable convolutional networks [49], proposed by Dai et al., and deformable skip-connections [50], proposed by Siarohin et al. Since the motivation for this work came from the expensiveness of sensors such as Lidar, we discourage the use of such sensors. However, using deep learning methods for estimating depth data is also an area of interest for future work.

6. Conclusion

In this paper, we presented an end-to-end method for translating egocentric views from RGB cameras such as those installed on vehicles into bird's eye views of the environment the subject vehicle was present in. One of the biggest hurdles is that egocentric views have a high level of distortion due to perspective, whereas a bird's eye view has a consistent scaling. The two are quite opposite in terms of geometric alignment. Previous traditional methods such as handcrafted homography transformations are not generalizable, and they do not work very well for views with minimal vertical leverage (e.g., view from the dashcam). More modern methods that use external sensors such as LIDAR can be very costly and computationally extensive. Taking all this into consideration, we develop our method to only use RGB frames from a single inexpensive camera installed in the car and so that it can be used for inference on the go on most modern mobile systems. We treat this as a task of view translation and implement it for two different use cases, one where we have a single image and one where we have a sequence of frames. We use an adversarial approach for training the model and experiment with image-to-image and video-to-video translation. The results from both experiments show that this can be a reliable approach to perform this task, and in the future, it can be used in the real world. However, there do exist some limitations, such as artefacts and loss of details over time, and we provide key insights for future researchers on how the performance and accuracy can be improved for this specific task. The work opens up new avenues for research on environment sensing in autonomous vehicles that only use dashcams as a sensor. While we have only shown the efficacy of this work for vehicle data, this can be extended to all sorts of egocentric views such as wearable cameras, and cameras installed on domestic assistant robots.

Data Availability

All data generated or analyzed during this study are included in this published article. Data is available at https://aimagelab.ing.unimore.it/imagelab/page.asp?IdPage=19.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

References

[1] Y. Jiang, F. Gao, and G. Xu, "Computer vision-based multiple-lane detection on straight road and in a curve," in *2010 International Conference on Image Analysis and Signal Processing*, pp. 114–117, Zhejiang, China, April 2010.

[2] K. Yu, L. Tan, S. Mumtaz et al., "Securing critical infrastructures: deep learning-based threat detection in the IIoT," *IEEE Communications Magazine*, 2021.

[3] K. Yu, Z. Guo, Y. Shen, W. Wang, J. C. Lin, and T. Sato, "Secure artificial intelligence of things for implicit group recommendations," *IEEE Internet of Things Journal*, 2021.

[4] A. Agarwal, C. V. Jawahar, and P. J. Narayanan, *A Survey of Planar Homography Estimation Techniques*, Centre for Visual Information Technology, Tech. Rep. IIIT/TR/2005/12, 2005.

[5] K. Yu, M. Arifuzzaman, Z. Wen, D. Zhang, and T. Sato, "A key management scheme for secure communications of information centric advanced metering infrastructure in smart grid," *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 8, pp. 2072–2085, 2015.

[6] L. Tan, H. Xiao, K. Yu, M. Aloqaily, and Y. Jararweh, "A blockchain-empowered crowdsourcing system for 5G-enabled smart cities," *Computer Standards Interfaces*, vol. 76, p. 103517, 2021.

[7] L. Tan, K. Yu, A. K. Bashir et al., "Toward real-time and efficient cardiovascular monitoring for COVID-19 patients by 5G-enabled wearable medical devices: a deep learning approach," *Neural Computing and Applications*, 2021.

[8] L. Tan, K. Yu, F. Ming, X. Chen, and G. Srivastava, "Secure and resilient artificial intelligence of things: a HoneyNet approach for threat detection and situational awareness," *IEEE Consumer Electronics Magazine*, p. 1, 2021.

[9] L. Tan, N. Shi, K. Yu, M. Aloqaily, and Y. Jararweh, "A blockchain-empowered access control framework for smart devices in green internet of things," *ACM Transactions on Internet Technology*, vol. 21, no. 3, pp. 1–20, 2021.

[10] V. Scinteie, *Autonomous vehicles and industry 5.0 are the basis for future-oriented traffic concepts in smart cities*, 2019.

[11] C. Feng, K. Yu, A. K. Bashir et al., "Efficient and secure data sharing for 5G flying drones: a blockchain-enabled approach," *IEEE Network*, vol. 35, no. 1, pp. 130–137, 2021.

[12] P. Jain and C. Jawahar, "Homography Estimation from Planar Contours," *IEEE*, pp. 877–884, 2006.

[13] X. Li, X. Fang, C. Wang, and W. Zhang, "Lane detection and tracking using a parallel-snake approach," *Journal of Intelligent Robotic Systems*, vol. 77, no. 3-4, pp. 597–609, 2015.

[14] I. S. Kholopov, "Bird's eye view transformation technique in photogrammetric problem of object size measuring at low-altitude photography," *Advances in Engineering Research*, vol. 133, 2017.

[15] A. Abbas and A. Zisserman, "A geometric approach to obtain a bird's eye view from an image," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pp. 4095–4104, Seoul, Korea (South), October 2019.

[16] H. Li, K. Yu, B. Liu, C. Feng, Z. Qin, and G. Srivastava, "An efficient ciphertext-policy weighted attribute-based encryption for the Internet of health things," *IEEE Journal of Biomedical and Health Informatics*, p. 1, 2021.

[17] H. Le, F. Liu, S. Zhang, and A. Agarwala, "Deep homography estimation for dynamic scenes," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7652–7661, Seattle, WA, USA, June 2020.

[18] S. A. Abbas and A. Zisserman, "A geometric approach to obtain a bird's eye view from an image," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, Seoul, Korea (South), October 2020.

[19] J. Zarzar, S. Giancola, and B. Ghanem, "Efficient bird eye view proposals for 3D Siamese tracking," 2019, http://arxiv.org/abs/1903.10168.

[20] A. E. Sallab, I. Sobh, M. Zahran, and N. Essam, "Lidar sensor modeling and data augmentation with GANs for autonomous driving," 2019, http://arxiv.org/abs/1905.07290.

[21] J. Zarzar, S. Giancola, and B. Ghanem, "Efficient tracking proposals using 2D-3d Siamese networks on Lidar," 2019, http://arxiv.org/abs/1903.10168.

[22] L. Zhen, A. K. Bashir, K. Yu, Y. D. al-Otaibi, C. H. Foh, and P. Xiao, "Energy-efficient random access for LEO satellite-assisted 6G Internet of remote things," *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5114–5128, 2021.

[23] L. Zhen, Y. Zhang, K. Yu, N. Kumar, A. Barnawi, and Y. Xie, "Early collision detection for massive random access in satellite-based Internet of Things," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 5, pp. 5184–5189, 2021.

[24] A. Palazzi, G. Borghi, D. Abati, S. Calderara, and R. Cucchiara, "Learning to map vehicles into bird's eye view," in *International Conference on Image Analysis and Processing*, pp. 233–243, Springer, 2017.

[25] K. Yu, L. Tan, L. Lin, X. Cheng, Z. Yi, and T. Sato, "Deep-learning-empowered breast cancer auxiliary diagnosis for 5GB remote E-health," *IEEE Wireless Communications*, vol. 28, no. 3, pp. 54–61, 2021.

[26] P. Isola, J.-Y. Zhu, T. Zhou, and A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5967–5976, Honolulu, 2017.

[27] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 172–189, Munich, Germany, 2018.

[28] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan, "Unsupervised pixel-level domain adaptation with generative adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 95–104, Honolulu, 2017.

[29] M.-Y. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," in *31st Conference on Neural Information Processing Systems (NIPS 2017)*, Long Beach, CA, USA, 2017.

[30] M. Y. Liu and O. Tuzel, "Coupled generative adversarial networks," *Advances in neural information processing systems*, vol. 29, pp. 469–477, 2016.

[31] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, "Learning from simulated and unsupervised images through adversarial training," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2107–2116, Hololulu, 2017.

[32] Y. Taigman, A. Polyak, and L. Wolf, "Unsupervised cross-domain image generation," *arXiv preprint arXiv*, vol. 1611, Article ID 02200, 2016.

[33] T. C. Wang, M. Y. Liu, J. Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional gans," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8798–8807, 2018.

[34] J.-Y. Zhu, T. Park, P. Isola, and A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232, 2017.

[35] J.-Y. Zhu, R. Zhang, D. Pathak et al., *Toward Multimodal Image-to-Image Translation*, 2017.

[36] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu et al., "Video-to-video synthesis," *arXiv preprint arXiv*, vol. 1808, Article ID 06601, 2018.

[37] Z. Guo, Y. Shen, A. K. Bashir, K. Yu, and J. C. W. Lin, "Graph embedding-based intelligent industrial decision for complex sewage treatment processes," *International Journal of Intelligent Systems*, 2021.

[38] I. Goodfellow, J. Pouget-Abadie, M. Mirza et al., "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds., pp. 2672–2680, Curran Associates, Inc., 2014.

[39] K. Yu, L. Lin, M. Alazab, L. Tan, and B. Gu, "Deep learning-based traffic safety solution for a mixture of autonomous and manual vehicles in a 5G-enabled intelligent transportation system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 4337–4347, 2021.

[40] Z. Guo, K. Yu, A. Jolfaei, A. K. Bashir, A. O. Almagrabi, and N. Kumar, "A fuzzy detection system for rumors through explainable adaptive learning," *IEEE Transactions on Fuzzy Systems*, p. 1, 2021.

[41] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, https://arxiv.org/abs/1411.1784.

[42] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," *ECCV*, 2016.

[43] Z. Guo, K. Yu, Y. Li, G. Srivastava, and J. C.-W. Lin, "Deep learning-embedded social Internet of Things for ambiguity-aware social recommendations," *IEEE Transactions on Network Science and Engineering*, 2021.

[44] M. Mathieu, C. Couprie, and Y. Lecun, *Deep Multi-Scale Video Prediction Beyond Mean Square Error*, 2015.

[45] Z. Guo, L. Tang, T. Guo, K. Yu, M. Alazab, and A. Shalaginov, "Deep graph neural network-based spammer detection under the perspective of heterogeneous cyberspace," *Future Generation Computer Systems*, vol. 117, pp. 205–218, 2021.

[46] X. Wang and A. Gupta, "Generative image modeling using style and structure adversarial networks," in *Computer Vision – ECCV 2016*, vol. 9908, pp. 318–335, Springer, 2016.

[47] K. Yu, L. Tan, M. Aloqaily, H. Yang, and Y. Jararweh, "Blockchain-enhanced data sharing with traceable and direct revocation in IIoT," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 11, pp. 7669–7678, 2021.

[48] O. Ronneberger, P. Fischer, and T. Brox, "U-net: convolutional networks for biomedical image segmentation," 2015, https://arxiv.org/abs/1505.04597.

[49] J. Dai, H. Qi, Y. Xiong et al., "Deformable convolutional networks," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 764–773, Venice, Italy, October 2017.

[50] A. Siarohin, E. Sangineto, S. Lathuilière, and N. Sebe, "Deformable GANs for pose-based human image generation," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3408–3416, Salt Lake City, UT, USA, June 2018.