Inverse Perspective Mapping generation on road vehicle images using Generative Adversial Networks

Victor Escribano Garcia, Alejandro Acosta Montilla, and Oriol Contreras Perez

Abstract—Inverse Perspective Mapping (IPM) is a crucial task in computer vision and autonomous navigation systems. Converting street photos to IPMs allows for accurate estimation of distances and improved scene understanding. In this paper, we propose a novel approach that leverages Generative Adversarial Networks (GANs) to transform street photos to IPMs. Our method combines the power of deep learning-based image synthesis with the ability to handle perspective distortions, enabling IPM generation from arbitrary street images. We demonstrate the effectiveness of our approach on a diverse set of real-world street scenes and experimental results show that our method achieves great performance in terms of visual quality and accuracy of the generated IPMs. Code available at https://github.com/VictorEscribano/IPM_GAN

Index Terms—Generative Adversarial Networks (GANs), Inverse Perspective Mapping (IPM), computer vision, image synthesis, street scene understanding.

1 Introduction

Inverse Perspective Mapping (IPM) is a fundamental task in computer vision and autonomous navigation systems. It involves transforming a perspective-distorted street photo into an Inverse Perspective Map, which provides a top-down view of the scene with accurate geometric properties. IPMs enable accurate distance estimation, object detection, and scene understanding, making them essential for applications such as autonomous driving, surveillance, and robotics.

Traditional approaches to IPM generation rely on explicit geometric transformations and assumptions about the scene. These methods require precise camera calibration, accurate estimation of camera parameters, and knowledge of the scene geometry. However, they are often sensitive to calibration errors and may not handle complex perspective distortions well. Additionally, these methods may not generalize to arbitrary street images, limiting their practicality in real-world scenarios.

2 State of the art

Recent advancements in deep learning, particularly the emergence of Generative Adversarial Networks (GANs), have revolutionized image synthesis tasks. GANs are composed of a generator network and a discriminator network, trained in an adversarial manner. The generator learns to synthesize realistic images, while the discriminator learns to distinguish between real and generated images. GANs have been successfully applied to various image synthesis tasks, including style transfer, image translation, and super-resolution.

In this study, we propose the application of a conditional Generative Adversarial Network (cGAN) based on the pix2pix [1] model to generate bird's eye view images from regular perspective views of the road. Our approach eliminates the need for explicit geometric assumptions and leverages the power of deep neural networks to learn the mapping between perspectives. We present our dataset creation process, including image extraction from the kitty dataset and homographic transformations, and describe our comprehensive evaluation metrics and experiments. The results demonstrate the effectiveness of our cGAN approach in generating realistic and accurate IPMs, contributing to the advancement of IPM generation for various applications, such as autonomous driving and scene understanding.

3 Methodology

Our proposed methodology utilizes Generative Adversarial Networks (GANs) for transforming street photos to Inverse Perspective Maps (IPMs). The overall pipeline consists of three main steps: data collection and preprocessing, GAN training, and IPM GAN inference.

3.1 Data Collection and Prepossessing:

The dataset used in this study is derived from the well-known kitty dataset, which consists of a collection of road scene images captured from a moving vehicle. From the kitty dataset [2], we extracted a total of 1.500 raw images specifically focusing on the road area. This subset of images serves as the basis for our training and evaluation process. By utilizing a subset of the kitty dataset, we ensure a diverse range of road scene perspectives and variations, enabling our model to learn robustly.

To complete the other half of the dataset, the corresponding IPM images from the raw road images from the kitty dataset need to be generated. As no dataset of this kind exists, we developed our own IPM calibration tool and automate the process of generating the remaining dataset.

This calibration tool is utilized to obtain the homographic matrix H through a intuitive GUI process developed

in tkinter [3]. This matrix captures the geometric mapping between the perspective view and the bird's eye view. By applying the homographic matrix to each perspective view image in the dataset, we are able to transform them into corresponding bird's eye view images. This transformation ensures accurate geometric properties and enables the synthesis of top-down road scene representations.

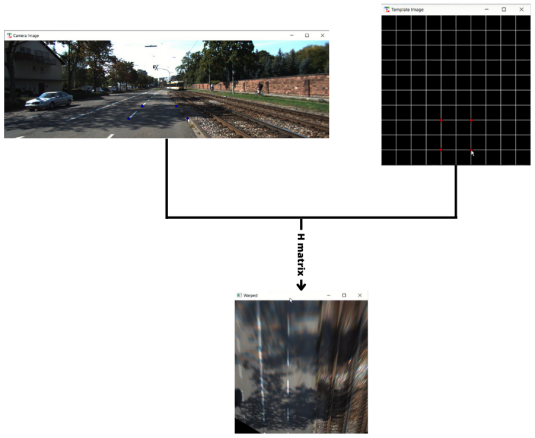

Fig. 1. Calibration process to obtain the Homography matrix, captures extracted from Calibration Tutorial YT video.

3.2 Conditional GAN:

The pix2pix model [1] serves as the foundation for our conditional GAN architecture. The pix2pix model is a popular choice for image-to-image translation tasks, and we adapt it to our specific cGAN setup for IPM generation.

3.2.1 The Generator (Unet):

The generator network in our cGAN takes a regular perspective view images of 128x128 pixels as input and aims to generate a corresponding bird's eye view image. It consists of an encoder-decoder architecture with skip connections (Unet), allowing the network to effectively capture and transfer the relevant information across different scales. The generator learns to transform the perspective view images into bird's eye view representations by leveraging the training data and the adversarial feedback from the discriminator. The encoder-decoder architecture consists of:

encoder:
C64-C128-C256-C512-C512-C512-C512

decoder:
CD512-CD512-CD512-C512-C256-C128

After the last layer in the decoder, a convolution is applied to map to the number of output channels (3), followed by a tanh function. As an exception to the above notation, BatchNorm is not applied to the first C64 layer in the encoder. All ReLUs in the encoder are leaky, with slope 0.2, while ReLUs in the decoder are not leaky.

The U-Net architecture is identical except with skip connections between each layer i in the encoder and layer $n-i$ in the decoder, where n is the total number of layers. The skip connections concatenate activations from layer i to layer $n-i$. This changes the number of channels in the decoder:

U-Net decoder:
CD512-CD1024-CD1024-C1024-C1024-C512

The Generator Loss

The generator loss in our cGAN model is a combination of the adversarial loss and the L1 loss. The adversarial loss is calculated using a sigmoid cross-entropy loss between the generated bird's eye view images and an array of ones, indicating the discriminator's goal of classifying the generated images as real. This loss encourages the generator to produce images that are more realistic and visually coherent.

In addition to the adversarial loss, as on the original pix2pix paper [1] we also incorporate the L1 loss, which measures the mean absolute error (MAE) between the generated bird's eye view image and the corresponding ground truth image. This loss encourages the generator to generate images that are structurally similar to the target images, promoting accurate geometric properties and scene representations in the generated bird's eye view.

To balance the importance of the adversarial and L1 losses, a weighting factor LAMBDA is applied. In our study, we opted for a lower value of LAMBDA in the generator loss to avoid producing blurry or desenfocated images. Higher LAMBDA values, as suggested in the pix2pix paper [1], prioritize pixel-level accuracy but can result in smoothing out important details and introducing blurriness. By reducing the weight on the L1 loss, we prioritize capturing structural characteristics and accurately representing the geometric properties of the road scenes in the generated bird's eye view images. This decision helps preserve important details and artifacts caused by distortion, which are crucial for IPM images.

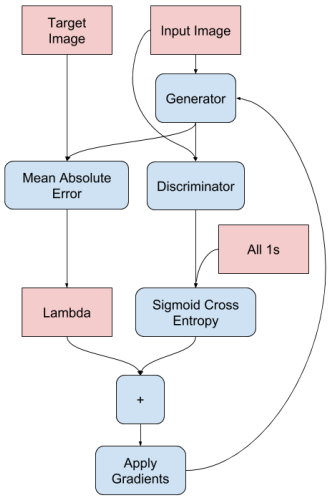

Fig. 2. Diagram of the generator loss pipeline.

3.2.2 The Markovian Discriminator (PatchGAN):

In the conditional Generative Adversarial Network (cGAN) architecture, the discriminator plays a crucial role in distinguishing between real and generated images. In the pix2pix model [1], the discriminator takes as input both the original image and the target image, and its task is to classify whether each image patch is real or fake. This approach is known as the PatchGAN classifier.

Each layer in the encoder is followed by a batch normalization layer (BatchNorm), except for the first layer, and all layers use leaky ReLU activations with a slope of 0.2. In our aproach we used only one downsample layer as descrived before and a last layer in the decoder with a sigmoid activation function to produce the probability map.

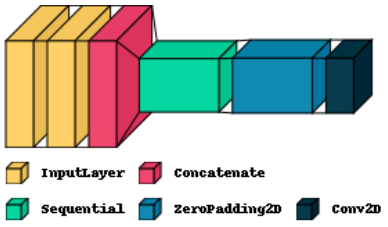

Fig. 3. Discriminator visualization with visuakeras library.

Normally, GANs discriminators returns true or false as output of the discrimination, in our Markov discriminator we have a matrix of 60x60 pixels indicating the range of trueness or falseness of each patch, to properly have a single value output of this discrimination we delimited this output between -1 and 1 with a *tanh* function, then computed te mean of all the patch and discriminate true images if the mean is above 0 or false if is bellow 0. On the metrics evaluation 3.4 section this accuracy evaluation process will be explained in detail.

The Discriminator Loss

The discriminator loss in our GAN model is calculated using sigmoid cross-entropy loss. For each input pair of an original image and its corresponding target image, the discriminator aims to classify each image patch as real or fake. The sigmoid cross-entropy loss measures the difference between the predicted probabilities and the ground truth labels.

To compute the discriminator loss, we compare the predicted probabilities for the patches in both the original image and the generated image with corresponding patches in the target image. The loss encourages the discriminator to correctly classify the patches as real or fake, thereby providing feedback to the generator to improve its ability to generate more realistic images.

The overall discriminator loss is calculated as the sum of the losses for the original image patches and the generated image patches. This adversarial loss encourages the discriminator to become more accurate in discriminating real and fake patches, leading to better training of the generator.

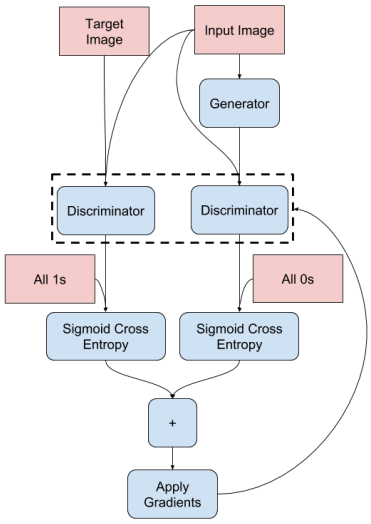

Fig. 4. Diagram of the discriminator loss pipeline.

3.3 Training Process:

3.3.1 Parameter tuning

As seen previously there are some parameters, such as Lambda, that need to be tuned in order to have the most optimal model during training. Some of these parameters are the learning rate, batch size and patchGAN size. The choice of batch size depends on various factors such as the available computational resources, the size of the dataset, and the model architecture. The original investigation [1] recommended a low batch size, in most of the cases this batch size was 1. In our approach, considering the available computational resources and the number of training iterations we have conducted, we have found that a batch size of 2 yields a good balance between training speed and performance. Pix2pix paper recommended a base learning rate for both, discriminator and generator, of $2 \cdot 10^{-4}$. To ensure that our model will perform okay on this value, the learning rate has been searched using the Learning Rate Finder method mentioned in [4]. This method consist in iteratively increasing the learning rate and observing the corresponding loss or metric values after 20 epochs. The learning rate is initially set to a small value, in this case 10^{-6}, and gradually increased in each iteration until 10^{0}.

The search has been performed separately for the generator and the discriminator (Figs. 5 and 6). It's important to note that the best learning rate is not necessarily the one with the lowest loss, but rather the one that exhibits the most rapid decrease in loss. This approach helps to identify the learning rate that allows for effective learning and convergence of the model must be around 10^{-6} and 10^{-5}, thus the selected final value for our training process is 2^{-5}.

Finally, as our dataset consists of images with larger structures and a global context is important, a larger patch size than the originally recomended (30x30 px) has been choosen, with a totall size of 66x66 px as menitoned on section 3.2.2.

Fig. 5. Learning Rate Finder method applied on the generator .

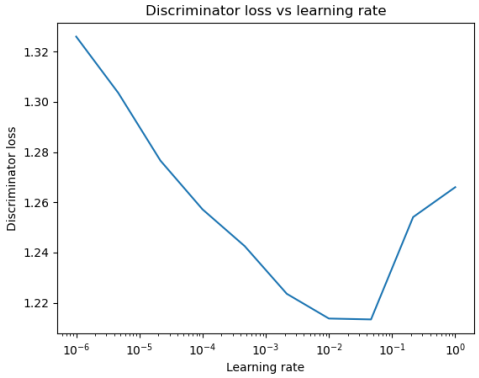

Fig. 6. Learning Rate Finder method applied on the discriminator .

3.3.2 Training results

During the training process, both the generator and discriminator components of the pix2pix model are optimized by minimizing specific loss functions. These loss functions play a crucial role in training the model and achieving the desired results.

During the training process, the generator and discriminator losses are optimized in an adversarial manner. The generator aims to minimize its own loss while simultaneously maximizing the discriminator's loss. Conversely, the discriminator aims to minimize its own loss by correctly classifying real and generated images.

As the training progresses, both the generator and discriminator losses should ideally decrease. The generator loss indicates the quality of the generated images, with a lower loss implying higher image fidelity. The discriminator loss reflects the discriminator's ability to distinguish between real and generated images, with a lower loss indicating improved discrimination performance.

The training results of the pix2pix model can be summarized as follows:

The loss of the generator (Fig. 7) has decreased steadily over the training iterations and has converged to a low value. This indicates that the generator has learned to generate output images that closely resemble the target images. The decreasing loss demonstrates the improvement in image quality and fidelity achieved by the generator during the training process.

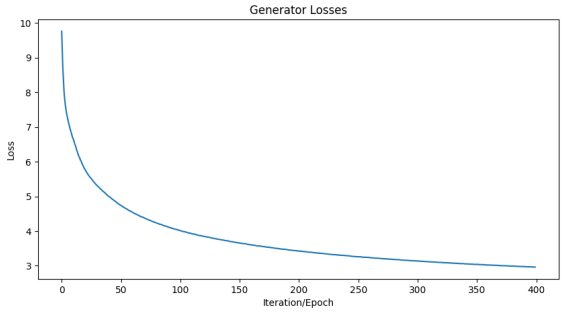

Fig. 7. Generator loss.

On the other hand, the loss of the discriminator (Fig. 8) initially decreased but then started to stabilize at a fixed value. It did not decrease as significantly as the generator's loss. However, it is important to note that the discriminator loss reaching a stable value does not necessarily indicate poor performance. It suggests that the discriminator has learned to accurately classify real and generated images and has achieved a level of discrimination competence.

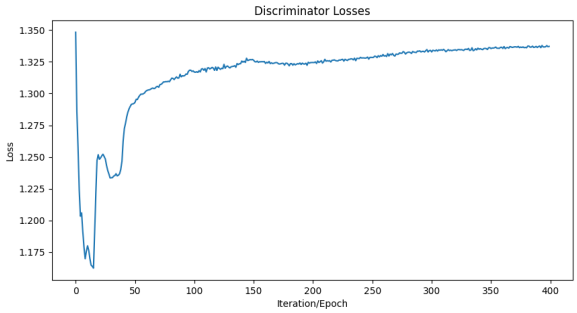

Fig. 8. Discriminator loss.

Despite the discriminator's loss not decreasing as much as the generator's loss, the accuracies of the discriminator are good and hover around 50% as will be explained on section 3.4. This is in line with the objective of the pix2pix model, which aims to train the discriminator to perform binary classification between real and generated images. Attaining accuracies around 50% indicates that the discriminator is making fair judgments and performing its classification task effectively.

3.4 Metrics evaluation:

The assessment of the results obtained from a GAN network is a hot topic that the current research is already tackling to ensure that in the future there would be suitable and reliable metrics for these image translation problems. Consequently, there is a lack of reliable measurements that can be used to determine objectively which is the best model in terms of images generated. Additionally, the discriminator used is a PatchGAN that yields as output a matrix of pixels likelihood of being real instead of yielding directly if the generated image is real or fake labelling it as 1 or 0 as commented in 3.2.2.

Hence, in terms of evaluation, since the GANs generate synthetic images from random noise or some input images, it is difficult to compare the degree of similarity that 2 images might have. However, two different metrics will be taken into consideration to determine the outperform of the different models proposed and to help in deciding which one is the best choice: the frechet inception distance, the discriminator real images accuracy and the discriminator fake images accuracy.

- **Frechet inception distance (FID):** It is a well - known approach mathematically used for assessing the minimum distance between 2 curves. However, it is used in GAN networks when synthesizing images and it is based on checking the similarities between 2 images computing and comparing the mean and the covariance of both images. Then, if the FID is being reduce, it means both images are more similar and that the network is learning to generate real images. This FID is computed following this equation:

$$FID^2 = (\mu_r - \mu_g)^2 + \Sigma_r + \Sigma_g - 2\sqrt{\Sigma_r \Sigma_g} \quad (1)$$

Where from 1 μ_r and Σ_r are the mean and covariance for the ground truth meanwhile μ_g and Σ_g are the ones for the generated image.

The literature normally address this indicator using a pre-trained network called InceptionV3 which receives as input the generated image and the ground truth and takes as output 2 arrays of size 1x2048 and from there all FID is computed. However, it is big time consuming and increases massively the time for an epoch to be completed. For this reason, this approach has been simplified flattening both real and generated images across all the 3 channels in one single array and then computing the mean and the covariance of each flattened image to find the FID between both. Consequently, since it is comparing the means and covariances of both images, the result of this simplified approach must yield similar results with small distances as long as both images are more similar.

- **Discriminator real images accuracy (DRIA):** It is the accuracy of the discriminator when classifying real images as real from the whole dataset computed as the total of images predicted as real divided by the total images on the dataset. From definition, it is clear this metric is analogous to the recall for the case of a confusion matrix on a real images vs fake image classification with the following equation:

$$DRIA = TP/(TP + FN) \quad (2)$$

- **Discriminator fake images accuracy (DFIA):** It is the accuracy of the discriminator when classifying fake images as real from all images generated computed as the total of images predicted as real divided by the total of images generated. From definition, it is clear this metric is analogous to the False Positive Rate for the case of a confusion matrix on a real image vs fake image classification with the following equation:

$$DFIA = FP/(FP + TN) \quad (3)$$

However, both metrics defined in equations 2 and 3 are called as accuracy for each image class since it is only being considered one image class (real or fake) at each metric, never mixing the other class as real value input. Moreover, their computation is not straightforward from the discriminator output since it is a patch-style discriminator and the post-processing steps depicted in Fig. 9 are used to yield the result for each image regardless of its type. Next, on one side, the DRIA value at the beginning of a

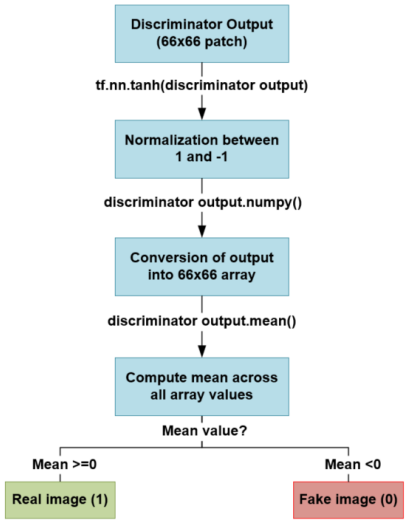

Fig. 9. Pipeline for discriminator output post-processing with each image.

training process must be really high and it must be reduced during the training procedure to converge to a value around 50%. On the other side, the DFIA must be really low at the beginning at it must converge to a value around 50% as well. All this means that the discriminator at the end is not able to distinguish between real and fake images.

Regarding the losses for the model, despite not being used as a metric, both are observed at all times in all models to ensure the training process is leading to a minimization of both losses and checking its convergence to some value. It is a good practice to make sure losses are being minimized since it is an indicator that both are learning and improving the GAN network performance. However, obtaining a continuous minimization of the losses does not lead to better image results and for this reason cannot be considered as metrics for the problem.

Finally, from all experimentation done, on this section it is only shown the results for the best model chosen from all trained. It must also be taken into consideration that regardless of the differences in accuracies and frechet distances obtained, the human evaluation on the images obtained is necessary since not the best models in terms of accuracy and frechet distances lead to the best image results. Moreover, the Figs. 10 and 11 portrays the metric results and its evolution along all epochs for the chosen model with image input size 128x128, 400 epochs, batch size 2, patch size 66x66 and learning rates of 0.00002 and $\beta_1 = 0.5$ and $\beta_2 = 1$ for both discriminator and generator networks. It is clear the GAN network from Fig.10 and 11 is learning but it will need more time to increase accuracy for the fake images up to 50%, which means more epochs and it will take computationally more time, but the system is stable

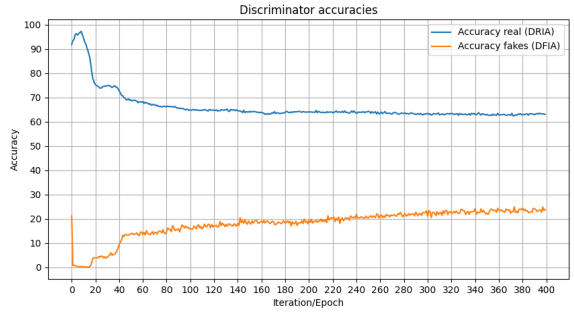

Fig. 10. Discriminator accuracies: DRIA vs DFIA along 400 epochs.

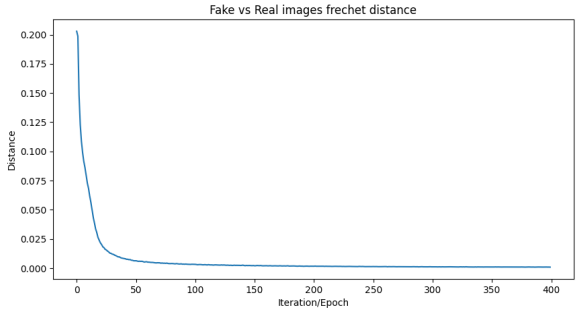

Fig. 11. Frechet inception distance (FID) along 400 epochs.

and converging to the desired values. However, Tables 1 and 2 with all features of all models and all metric values for all models trained with the hyperparameters used on them can be found on the Appendix for more comparison.

4 Obtained Results

The obtained results demonstrate a mix of successful and challenging outcomes. Some images show highly satisfactory results (for example Figs. 12 and 13), closely resembling the desired Inverse Perspective Images (IPM).

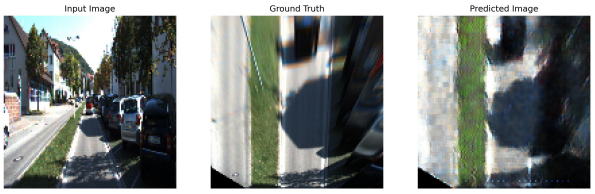

Fig. 12. Evaluation of Generated images from the test dataset.

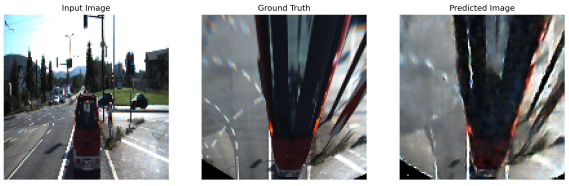

Fig. 13. Evaluation of Generated images from the test dataset.

However, there are some images (for example Figs. 14 and 15) that exhibit less desirable and confusing results.

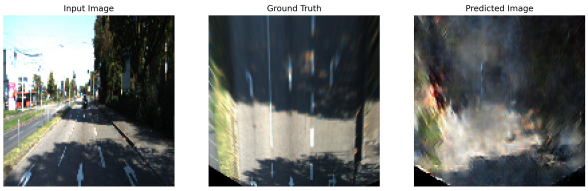

Fig. 14. Evaluation of Generated images from the test dataset.

Fig. 15. Evaluation of Generated images from the test dataset.

One of the main reasons behind these less favorable outcomes is believed to be overfitting. Despite efforts to mitigate overfitting by filtering out repeated images, applying augmentations, and ensuring dataset diversity, the limited size of the dataset remains a significant constraint. With only 1500 images, it is relatively small considering the broad application we aim to address. Consequently, this limitation contributes to difficulties encountered when generating IPM for certain images in the testing dataset.

While the model has demonstrated promising capabilities, it is important to acknowledge the impact of dataset size on performance. Future improvements should focus on expanding the dataset to include a more extensive range of scenarios and a greater quantity of images. This expansion would help overcome overfitting and enhance the model's generalization and performance on a wider range of testing images.

Overall, the obtained results highlight both the potential and limitations of the current model. Addressing the challenges associated with dataset size and overfitting will be crucial in further improving the performance and applicability of the model for the desired application of generating accurate and reliable Inverse Perspective Images.

5 Conclusion

In conclusion, applying conditional GANs to generate Inverse Perspective Images (IPM) has yielded highly satisfactory results. However, there are several avenues for future work to further improve the model.

Firstly, expanding the dataset with a richer variety of images from different scenarios would enhance the model's ability to generalize to a wider range of input conditions. Additionally, increasing the dataset size would provide more training samples and potentially improve the model's performance.

It would also be valuable to conduct a comprehensive study on how different values of Lambda, in combination

with various learning rates, impact the model's training and overall performance. Finding the optimal combination of Lambda and learning rate can potentially enhance the convergence and stability of the model.

Furthermore, it is crucial to highlight that the ultimate goal is to find a 3x3 matrix that can accurately transform images into IPM. Hence, a future direction could involve narrowing down the generator's task to learning this specific 3x3 matrix. By focusing on this transformation matrix, the GAN can be trained to effectively learn and find the optimal parameters to achieve the desired IPM generation.

In summary, while the application of conditional GANs for IPM generation has shown highly satisfactory results, future work should consider enriching the dataset, exploring the effects of different Lambda values and learning rates, and ultimately focusing on optimizing the generator to find the 3x3 matrix that performs the IPM transformation. These advancements would contribute to further enhancing the model's performance and its applicability to real-world scenarios.

6 Code

All codes provided for the development of this project can be found on the following GitHub repository: https://github.com/VictorEscribano/IPM_GAN

Appendix A

On the appendices will be gathered on 2 Tables the results from several iterations performed with different models when attempting to tune and decide whether could be the best ones. Note that all these training has been performed to decide which could be the best model and to decide the best values for all hyperparameters based on the metrics obtained and the visual inspection of the generated training and test images.

TABLE 1
Iterations matrix

Iter	Epochs	Batch	LR.Gen	LR.Dis	λ	Patch Size
01	400	4	2,00E-05	2,00E-05	20	32x32
02	400	4	2,00E-05	2,00E-05	20	64x64
03	400	2	2,00E-05	2,00E-05	20	64x64
04	400	2	2,00E-05	2,00E-05	5	64x64
05	400	2	2,00E-05	2,00E-05	1	64x64
06	400	2	2,00E-02	2,00E-02	20	64x64
07	400	2	2,00E-04	2,00E-04	20	64x64
08	400	2	2,00E-06	2,00E-06	20	64x64
09	400	2	2,00E-03	2,00E-03	20	64x64
10	400	8	2,00E-05	2,00E-05	20	64x64
11	400	2	2,00E-05	2,00E-05	50	64x64
12	400	10	2,00E-05	2,00E-05	20	64x64
13	400	4	2,00E-05	2,00E-05	20	64x64

TABLE 2
Accuracies from the model for the different iterations trained

Iter	Epochs	Batch	DRIA [%]	DFIA [%]
01	400	4	100	0
02	400	4	71.57	21.07
03	400	2	63.04	24.76
04	400	2	70.74	24.58
05	400	2	63.54	28.09
06	400	2	100	0
07	400	2	74.49	19.31
08	400	2	69.90	8.86
09	400	2	72.82	22.24
10	400	8	68.81	15.21
11	400	2	TBD	TBD
12	400	10	75.94	12.82
13	400	4	73.11	14.99

References

1. ALL, Phillip Isola et. *Image-to-Image Translation with Conditional Adversarial Networks*. Available also from: https://arxiv.org/pdf/1611.07004.pdf.
2. FRITSCH, Jannik; KUEHNL, Tobias; GEIGER, Andreas. A New Performance Measure and Evaluation Benchmark for Road Detection Algorithms. In: *International Conference on Intelligent Transportation Systems (ITSC)*. 2013.
3. ESCRIBANO, Victor. *Generating Bird's Eye View Images Using cGAN and IPM*. Available also from: https://www.youtube.com/watch?v=FLBrwnHSxnk.
4. SMITH, Leslie N. *Cyclical Learning Rates for Training Neural Networks*. Available also from: https://arxiv.org/pdf/1506.01186.pdf.
5. IANHOLING. *faceGAN$_{f}ace_{e}ditor$*. Available also from: https://github.com/ianholing/faceGAN_face_editor / blob / master / CariGAN_Pix2Pix_v1_from_walter.ipynb.
6. HMARTELB. *Pix2Pix-Timbre-Transfer*. Available also from: https://github.com/hmartelb/Pix2Pix-Timbre-Transfer/tree/master.
7. SURMENOK, Pavel. *Estimating an Optimal Learning Rate For a Deep Neural Network*. Available also from: https://towardsdatascience.com/estimating-optimal-learning - rate - for - a - deep - neural - network - ce32f2556ce0.
8. GIRO, Xavier. *T13 Generative Adversarial Networks GAN - Xavier Giro - UPC TelecomBCN Barcelona 2020*. Available also from: https://www.youtube.com/watch?v=PflmLttr6GU&ab_channel=ImageProcessingGroup-UPC%2FBarcelonaTECH.
9. BROWNLEE, Jason. *How to Implement the Frechet Inception Distance (FID) for Evaluating GANs*. Available also from: https://machinelearningmastery.com/how-to-implement-the-frechet-inception-distance-fid-from-scratch/.
10. TAN, Daryl. *A Hands-On Application of Homography: IPM*. Available also from: https://towardsdatascience.com/a-hands-on-application-of-homography-ipm-18d9e47c152f.