Tree-based Optimization for Image-to-Image Translation with Imbalanced Datasets on the Edge

Pasquale Coscia*, Angelo Genovese*, Vincenzo Piuri*, Francesco Rundo†, Fabio Scotti*
*Department of Computer Science, Università degli Studi di Milano, Italy
{*pasquale.coscia, angelo.genovese, vincenzo.piuri, fabio.scotti*}*@unimi.it*
†STMicroelectronics, ADG Central R&D, Catania, Italy
francesco.rundo@st.com

***Abstract*—Image-to-image (I2I) translation models typically refer to a class of adversarial architectures aiming to transfer an image content from a source domain to a target domain. To increase the image quality, data augmentation techniques or collecting new samples represent valid options yet lack of diversity and overfitting may negatively impact on the final results. In this regard, several practical scenarios do not permit to include new samples, or to employ powerful hardware, due to privacy policies or insufficient financial resources, leading to use imbalanced sets of images and favoring the more populated domain. To overcome these issues, we propose a simple and effective procedure to take advantage of the combination of critical learning parameters and demonstrate that averaging weights of multiple pre-trained I2I models is beneficial for increasing model performance, which can be optimized for edge computing without hurting the quality of synthesized images. To this end, we define a tree-based structure, including multiple I2I translation models, that outputs a single and more reliable network. We demonstrate that this strategy increases image quality and also show that our binary-tree learning procedure has a beneficial impact on edge devices, and it can be easily applied to architectures trained on different domains.**

***Index Terms*—Image-to-image translation, fine-tuning, edge devices, generative adversarial networks, optimization.**

I. Introduction

Generative methods for image manipulation represent a key-component of several computer vision areas due to their powerful impact on real-world applications [1]. Remarkable results of state-of-the-art models [2]–[4] demonstrate their ability to learn both local and global characteristics. To generate synthetic samples, adversarial training is widely employed with different types of objective functions to increase quality perception. In this respect, image translation aims at converting the content of an image from a source to a target domain. Early approaches [5], [6] were defined to consider only one domain, yet they were rapidly extended to multiple domains [7], [8] mainly to include different styles. Image-to-image (I2I) translation techniques are applied for increasing scene perception of autonomous vehicles, processing satellite images or segment medical images. Nevertheless, these models demand large-scale datasets for achieving satisfactory quality, and their training on low-power devices is impractical. One major limitation in applying I2I models for synthesizing realistic images in specific contexts, *e.g.*, industrial domain for defect generation, or style transfer for paintings, is, in fact, represented by the limited samples that can be collected leading to imbalanced domains. Such imbalance may introduce biases during training and lead to poor image translation quality. To overcome these problems, pre-trained models on large-scale datasets represent a valid alternative for different downstream tasks. For example, fine-tuning from multiple models has the advantage to increase the performance using an appropriate ensemble modeling and creates more robust architectures to domain shifts and out-of-distribution samples. Differently from classification tasks, where multiple predictions can be easily combined using their logits to output more confident results, image generation tasks are typically based on adversarial architectures, making it hard using pre-trained models. Several works address this important limitation in different ways, *e.g.*, via features interpolation [9] or creating synthesis networks as generative priors [10]. For image classification, Wortsman *et al.* [11] also improve out-of-distribution and zero-shot performance on different downstream tasks.

On the other hand, current advances in computer vision research have enabled safe and secure applications which can be more responsive if their inference is carried out on edge devices to prevent data leakage or because network connections to data centers may introduce unavoidable lags. Several industrial scenarios, in fact, demand to use these devices for optimizing some tasks (e.g., instrumentation control or image classification). Nevertheless, edge devices typically offer limited computational capabilities, requiring ad-hoc optimization procedures to reach performance similar to more powerful hardware or cloud services. Lowering weights precision of pre-trained networks from 32-bit floating point numbers (FP32) to 8-bit integers (INT8) is a typical optimization step used to save resource requirements and avoid privacy issues, but not sufficient to preserve optimal performance.

To train an accurate I2I translation model [12] in case of imbalanced datasets, we propose a simple and effective tree-based procedure able to take advantage of multiple available pre-trained models. Each node of this tree represents a model

This work was supported in part by the EC under project EdgeAI (101097300), by the Italian MUR under PON project GLEAN, and by project SERICS (PE00000014) under the MUR NRRP funded by the EU - NextGenerationEU. Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the Italian MUR. Neither the European Union nor Italian MUR can be held responsible for them.

trained on large-scale or similar datasets; two models are then coupled to fuse their weights and fine-tuned on the dataset of interest. This procedure is repeated at each level of the tree to converge to the best model used at inference time. We validate our approach simulating different degrees of imbalance that are typically met in practical scenarios, and propose several variants to only consider specific network layers. We also simulate a quantization procedure to train our model on an 8-bit device.

II. Related Work

Image-to-image translation. Generative adversarial networks (GANs) represent a powerful class of generative methods able to produce high-quality images and videos, and also the main methodology employed for I2I translation [13]. Isola *et al*. [6] define a conditional GAN for paired image-to-image translation while Zhu *et al*. [12] investigate the unpaired case introducing a cycle-consistency loss able to reconstruct the input image from the source domain. These works have been extended for many tasks using ad-hoc models (*e.g*., super-resolution [14], conditional image synthesis [15] and unsupervised learning [16]). Several approaches adopt contrastive learning for obtaining more useful representations of the content of images, and to solve the data imbalance problem, exploiting relations between positive and negative pairs [17], [18]. Cao *et al*. [9], instead, propose an interpolation at features level and impose a properly defined perceptual loss for estimating unknown image targets.

Fine-tuning. Fine-tuning typically involves a number of models trained using different hyperparameters, or large-scale datasets, and selects the model with the best performance on a held-out validation set. Several approaches to leverage pre-trained models as generative prior are proposed for conditional image generation, modification and restoration [19], [20]. Grigoryev *et al*. [21] investigate the usefulness of pre-trained generators and discriminators on large-scale datasets while Karris *et al*. [22] analyze low-data regimes proposing an adaptive discriminator augmentation technique for training stabilization. Mo *et al*. [23] demonstrate that fine-tuning GANs without modifying lower layers of the discriminator positively affects their learning process.

Edge devices. Edge devices provide several benefits in terms of privacy and scalability [24], [25]. Nevertheless, intensive computation and low-latency requirements of deep learning models cannot be met without proper optimization procedures. Pruning or fusion schemes are commonly adopted for faster computation [26], as well as reducing floating-point representation down to 16 or 8 bits [27]–[29]. Several libraries also propose efficient representations and compression schemes for edge hardware [30].

III. Method

I2I models represent a class of generative architectures able to synthesize images related to a target domain. To increase the quality of images synthesized by I2I models in case of imbalanced domains, we propose multiple fine-tuning steps that can be iteratively applied, focusing on critical learning parameters. In the following, we firstly describe an approach to increase perceptual performance on imbalanced datasets and then present an optimization procedure.

Problem formulation. Our aim is to translate an image $\mathbf{x} \in \mathbb{R}^{H \times W \times C}$ from an input domain X to an image $\mathbf{y} \in \mathbb{R}^{H \times W \times C}$ from an output domain Y. We are given two sets of unpaired instances, $\mathcal{X} = \{\mathbf{x} \in X\}$ and $\mathcal{Y} = \{\mathbf{y} \in Y\}$, composing a dataset $\mathcal{D}$, where $|\mathcal{X}| >> |\mathcal{Y}|$ ($|\cdot|$ denotes the number of images of the corresponding set). To solve our task, let M_1 and M_2 denote two models, trained to minimize an objective function $\mathcal{L}_1$ and $\mathcal{L}_2$, respectively, able to synthesize new images. For the sake of simplicity, we assume that M_1 and M_2 represent the same model containing L layers. Let $\mathbf{w}^i$ be the vector of weights for the i^{th} layer. The output of each network is obtained as follows:

$$f_w^M(\mathbf{x}) = \Phi_L \circ \Phi_{L-1} \circ ... \circ \Phi_1(\mathbf{x}), \tag{1}$$

where $\mathbf{x} \in \mathcal{X}$ represents the input, $\Phi(\cdot)$ is a function (*e.g*., a convolution or linear layer) and $\circ$ denotes the composition operator.

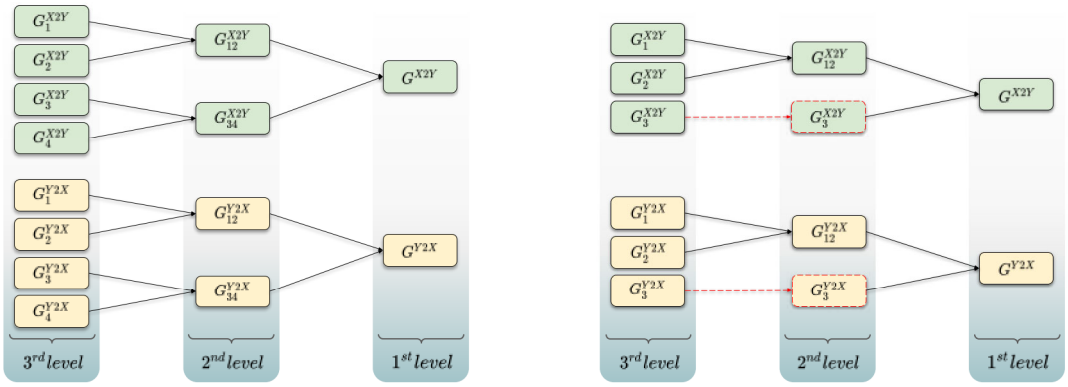
To combine the weights of the i^{th} layer of these networks, we consider a weighted average as follows:

$$\mathbf{w}_{\mathcal{D}}^i = \beta \cdot \mathbf{w}_{\mathcal{D}_1}^{i,M_1} + (1-\beta) \cdot \mathbf{w}_{\mathcal{D}_2}^{i,M_2}. \tag{2}$$

Here, $\mathbf{w}_{\mathcal{D}_1}^{i,M_1}$ and $\mathbf{w}_{\mathcal{D}_2}^{i,M_2}$ denote the weights of M_1 and M_2 previously trained on the datasets $\mathcal{D}_1$ and $\mathcal{D}_2$, respectively, while $\beta \in [0,1]$ represents the interpolation parameter defining the *importance* of a model compared to the other one. The trivial case with $\beta = 0$ (or $\beta = 1$) would simply correspond to a copy of one of the two models. Our procedure can be represented using a binary tree structure (see Fig. 1). More specifically, each node acts as a model (or subnetwork) that is obtained as a combination of two nodes from a previous level. Once the weights are combined, a fine-tuning step on $\mathcal{D}$ is performed to adapt them to the specific task. Let N denote the number of levels. The total number of nodes can be obtained as $2^N - 1$. Once this procedure is applied to all the pairs of models at the n^{th} level, it is repeated at the $(n-1)^{th}$ level, containing half of the models of the previous level. In case of an even number of models, two nodes can be randomly chosen, combined and fine-tuned. This represents a symmetric binary tree. By contrast, an odd number of models does not allow a combination of two arbitrary networks; in this case, one model can be fine-tuned and then "copied" to the next level. This represents an asymmetric binary tree.

Since I2I models typically involve multiple generators and discriminators as well as multiple encoders and decoders, this step can be applied to each subnetwork. Depending on the number of layers to use in this process, multiple fine-tuning strategies can be considered:

- All the weights of M_1 and M_2 are combined, as reported in Eq. 2, and then fine-tuned on $\mathcal{D}$ (S1);
- Only the weights of specific layers (for example, layers related to low-level features) of M_1 and M_2 are combined



(a) Symmetric

(b) Asymmetric

Fig. 1: Our strategy involves a binary-tree structure using images from two domains ($\mathcal{X}$ and $\mathcal{Y}$) collected in a dataset $\mathcal{D}$. Considering a model containing two generators, G^{X2Y} and G^{Y2X}, we firstly apply, at each level, a weighted average on the weights of the two models from the previous level and then fine-tune on $\mathcal{D}$ each combined model using fixed hyperparameters selected a-priori. If an odd number of models is available (b), one model is randomly selected, fine-tuned on $\mathcal{D}$, and then "copied" to the next level to be merged with another one. Likewise, this strategy can be applied to the weights of subnetworks or parts of the models.

and fine-tuned on $\mathcal{D}$ while the remaining part of the network is trained from scratch (S2-A);

- Only the weights of specific layers (for example, layers related to low-level features) of M_1 and M_2 are combined and *not* fine-tuned (*i.e.*, freezed) while the remaining part of the network is trained from scratch (S2-B);
- Specific subnetworks of M_1 and M_2 can mix the above strategies (S3);

Given the large availability of pre-trained architectures on large-scale datasets, this procedure can be easily extended to models trained on different data and potentially applied to multiple combinations of parameters that may affect the training process (*e.g.*, learning rate, optimizer or batch size).

Imbalance-aware loss. I2I models typically permit both a direct and inverse mapping, *i.e.*, $\mathcal{X} \rightarrow \mathcal{Y}$ and $\mathcal{Y} \rightarrow \mathcal{X}$. Nevertheless, the objective function may combine the training samples from both domains and, in case of imbalanced data, one set would certainly count more than the other limiting the model to focus on relevant features of the less populated domain. To increase the awareness towards these images during the training stage, assuming a loss function defined as $\mathcal{L} = \mathcal{L}^{\mathcal{X}\rightarrow\mathcal{Y}} + \mathcal{L}^{\mathcal{Y}\rightarrow\mathcal{X}}$, we consider a weighted combination as follows:

$$\mathcal{L}_{aware} = \lambda_{\mathcal{I}}\mathcal{L}^{X\rightarrow Y} + (1-\lambda_{\mathcal{I}})\mathcal{L}^{Y\rightarrow X}, \tag{3}$$

where $\lambda_{\mathcal{I}} \in [0,1]$ refers to the imbalance degree between the two domains. In this way, operations on images from $\mathcal{X}$ count more than operations on images from $\mathcal{Y}$.

Optimization. We also experiment the deployment of our models on edge devices using the Neural Network Compression Framework (NNCF) of the OpenVINO™ toolkit for 8-bit quantization in PyTorch. We consider the quantization-aware training (QAT) which is more robust to quantization after training. This process simulates a quantization procedure during training in order to treat the models as 8-bit networks at inference time.

	2-levels	3-levels	η
Model 1 (Very Slow)	✓	✓	0.00005
Model 2 (Slow)	✗	✓	0.00015
Model 3 (Fast)	✓	✓	0.0015
Model 4 (Very Fast)	✗	✓	0.030

TABLE I: Learning rates used with a 2-levels/3-levels binary tree.

Dataset	**Imbalance** (%)	$\lvert\mathcal{X}\rvert$	$\lvert\mathcal{Y}\rvert$
Apple → Orange	90	995	100
	95		50
	99		10
Orange → Apple	90	1,019	102
	95		51
	99		10
Horse → Zebra	90	1,067	107
	95		53
	99		11
Zebra → Horse	90	1,334	133
	95		67
	99		13

TABLE II: Number of training images per domain for each considered imbalance percentage.

IV. Results

To measure the impact on quality metrics when imbalanced domains are used, we compute the imbalance percentage $\mathcal{I}$ as $(|\mathcal{X}| - |\mathcal{Y}|)/|\mathcal{X}| \cdot 100$. The bigger this value, the greater the imbalance between the two domains is. For example, considering two domains including $1,000$ images each, *i.e.*, $|\mathcal{X}| = |\mathcal{Y}| = 1,000$, if the imbalance is set to 90%, we use $1,000$ images from domain $\mathcal{X}$ and 100 images from domain $\mathcal{Y}$.

		Apple → Orange		Orange → Apple		Horse → Zebra		Zebra → Horse	
Imbalance (%)	Tree levels	FID (↓)	LPIPS (↑)	FID (↓)	LPIPS (↑)	FID (↓)	LPIPS (↑)	FID (↓)	LPIPS (↑)
90	-	2.44	**0.65**	3.55	**0.67**	4.14	**0.63**	**2.29**	0.60
	2	2.19	**0.65**	2.15	0.65	10.03	0.61	3.13	**0.62**
	3	**1.72**	0.64	**1.95**	0.65	**3.66**	0.61	3.54	**0.62**
95	-	3.66	**0.65**	3.86	**0.66**	**5.11**	**0.64**	3.40	0.59
	2	3.88	**0.65**	**2.26**	0.65	7.21	0.60	3.54	**0.63**
	3	**3.25**	0.64	5.07	**0.66**	7.12	0.63	**3.38**	0.61
99	-	3.06	**0.65**	6.67	**0.65**	**3.76**	0.60	**4.05**	0.61
	2	**2.19**	**0.65**	**3.46**	0.63	4.88	**0.61**	6.20	0.61
	3	11.55	0.64	3.92	0.63	5.70	0.60	9.47	**0.64**

TABLE III: FID and LPIPS metrics. The first row of each subset refers to the Cycle-GAN architecture without applying our fine-tuning strategies. Images involving apples and oranges result more simple to be translated compared to images showing horses or zebras. S1 strategy is employed for both generators and discriminators.

		Apple → Orange		Orange → Apple		Horse → Zebra		Zebra → Horse	
Imbalance (%)	Tree levels	FID (↓)	LPIPS (↑)	FID (↓)	LPIPS (↑)	FID (↓)	LPIPS (↑)	FID (↓)	LPIPS (↑)
95	-	3.66	**0.65**	3.86	0.66	**5.11**	0.64	**3.40**	0.59
	2	**2.15**	0.64	2.40	0.66	6.23	**0.65**	3.92	0.61
	3	2.66	**0.65**	**2.31**	**0.67**	7.15	0.62	7.37	**0.64**

TABLE IV: FID and LPIPS metrics when only discriminators are retained from the previous layer and combined, while generators are trained from scratch at each level. S2-A strategy is employed.

			Apple → Orange	
Imbalance (%)	Tree levels	$\mathcal{L}_{aware}$	FID (↓)	LPIPS (↑)
95	-	✗	3.66	0.65
	-	✓	**3.50**	**0.66**
	2	✗	3.88	0.65
	2	✓	**3.24**	**0.66**
	3	✗	3.25	**0.64**
	3	✓	**2.51**	**0.64**

TABLE V: FID and LPIPS metrics for a single translation using a symmetric tree and an imbalance-aware loss. S1 strategy is employed for both generators and discriminators.

		Apple → Orange	
Imbalance (%)	Tree levels	FID (↓)	LPIPS (↑)
95	-	3.66	0.65
	3	**3.50**	**0.66**

TABLE VI: FID and LPIPS metrics for a single translation using an asymmetric tree with 3 models: very slow, fast and a model trained on the Horse → Zebra transformation. The last one is copied to the next level. S1 strategy is employed for both generators and discriminators.

			Apple → Orange	
Imbalance (%)	QAT	Epochs	FID (↓)	LPIPS (↑)
95	✗	-	**3.66**	0.65
	✓	2	4.47	**0.66**
	✓	5	4.45	**0.66**

TABLE VII: FID and LPIPS metrics obtained training Cycle-GAN for 50 epochs and then applying our quantization procedure (*i.e.*, transformation to 8 bits and fine-tuning for 2 and 5 epochs, respectively).

As baseline architecture, we use Cycle-GAN [12], a residual-based I2I translation model involving two couples of generator-discriminator, one for each mapping. It employs a direct and an inverse mapping, and introduces a cycle-consistency loss to constrain the learned transformation. An identity loss is used to avoid unnecessary modifications to the input images, and an adversarial loss is used to generate realistic outputs. Its final objective function can be represented as follows:

$$\mathcal{L} = \lambda \mathcal{L}_{GAN} + \lambda_{cycle} \mathcal{L}_{cycle} + \lambda_{id} \mathcal{L}_{id}. \tag{4}$$

We refer the reader to Zhu *et al.* [5] for a more detailed description of this architecture. We set β to 0.5 in Eq. 2, and use $\lambda = 1$, $\lambda_{cycle} = 5$ and $\lambda_{id} = 10$ for our fine-tuning steps. The optimizer is Adam and the learning rate η is set to 0.0002. At each level, we consider 50 epochs.

For our experiments, we adopt two datasets: Apple → Orange and Horse → Zebra. Statistics of both datasets are reported in Table II. To quantitative evaluate our approaches, we use two metrics: FID (Fréchet Inception Distance) [31] and LPIPS (Learned Perceptual Image Patch Similarity) [32] metric. We compute the former on features with dimensionality 64 (*i.e.*, first max pooling features).

In Table III, we evaluate our approach with a symmetric binary tree with 2 and 3 levels and different imbalances. The models are trained using the parameters reported in Table I. We note that increasing the number of levels has a beneficial impact on the learned features, in most cases. Images containing more complex characteristics (*e.g.*, zebras or horses), instead, do not appear much affected by our procedure, demonstrating that over-fitting cannot be easily solved. Furthermore, we show our qualitative results in Fig. 2. We observe a more consistent color structure for our synthesized images after several fine-

tuning steps. Similarly to Mo *et al*. [23], we report in Table IV an experiment without inheriting generators from previous layers. Fig. 3 also confirms the positive impact of our strategy. In Table V and Table VI we report the effect of our imbalance-aware loss and an asymmetric binary tree using a pre-trained model on a different dataset. Finally, to test the impact of quantization on our models, we first transform the original pre-trained FP32 model to INT8, and then, use fine-tuning for a number of epochs varying from 2 to 5. Table VII and Fig. 4 show that fine-tuning for deployment is a crucial step for limiting inevitable drops in performance.

V. Limitations

A major limitation of our approach is represented by the number of fine-tuning steps, which increases exponentially. Our techniques are, in fact, tested with a limited number of epochs and simulating an 8-bit training device. An imbalance-aware loss with fixed weights may also not properly focus on simple or hard samples to be learned. Finally, to obtain more robust results, multiple runs may be considered.

VI. Conclusion

To overcome overfitting and increase diversity for unpaired image-to-image translation models, we propose multiple fine-tuning strategies able to increase image quality for imbalanced domains. We demonstrate that a binary-tree structure can be employed when multiple generators and discriminators are involved. Our future work will be towards a more light-weighted procedure to reduce a drop in performance highlighted by specific datasets and images containing complex patterns, and a further detailed analysis comprising more learning parameters.

References

[1] S. Bond-Taylor, A. Leach, Y. Long, and C. G. Willcocks, "Deep generative modelling: A comparative review of vaes, gans, normalizing flows, energy-based and autoregressive models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[2] S. Mo, Z. Sun, and C. Li, "Representation disentanglement in generative models with contrastive learning," in *Proc. of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2023.

[3] Y. Xue, Y. Li, K. K. Singh, and Y. J. Lee, "Giraffe hd: A high-resolution 3d-aware generative model," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[4] T. Chen, Y. Zhang, X. Huo, S. Wu, Y. Xu, and H. S. Wong, "Sphericgan: Semi-supervised hyper-spherical generative adversarial networks for fine-grained image synthesis," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[5] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2017.

[6] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[7] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[8] Y. Choi, Y. Uh, J. Yoo, and J.-W. Ha, "Stargan v2: Diverse image synthesis for multiple domains," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[9] J. Cao, L. Hou, M.-H. Yang, R. He, and Z. Sun, "Remix: Towards image-to-image translation with limited data," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[10] T. Wang, T. Zhang, B. Zhang, H. Ouyang, D. Chen, Q. Chen, and F. Wen, "Pretraining is all you need for image-to-image translation," *arXiv*, 2022.

[11] M. Wortsman, G. Ilharco, S. Y. Gadre, R. Roelofs, R. Gontijo-Lopes, A. S. Morcos, H. Namkoong, A. Farhadi, Y. Carmon, S. Kornblith, and L. Schmidt, "Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time," in *Proc. of the International Conference on Machine Learning (ICML)*, 2022.

[12] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2017.

[13] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[14] 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 *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[15] Y. Li, X. Chen, F. Wu, and Z.-J. Zha, "Linestofacephoto: Face photo generation from lines with conditional self-attention generative adversarial networks," in *Proc. of the ACM International Conference on Multimedia*, 2019.

[16] M.-Y. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," in *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[17] J. Zhu, S. Tang, D. Chen, S. Yu, Y. Liu, M. Rong, A. Yang, and X. Wang, "Complementary relation contrastive distillation," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[18] T. Park, A. A. Efros, R. Zhang, and J.-Y. Zhu, "Contrastive learning for unpaired image-to-image translation," in *Proc. of the IEEE/CVF European Conference on Computer Vision (ECCV)*, 2020.

[19] X. Pan, X. Zhan, B. Dai, D. Lin, C. C. Loy, and P. Luo, "Exploiting deep generative prior for versatile image restoration and manipulation," in *Proc. of the IEEE/CVF European Conference on Computer Vision (ECCV)*, 2020.

[20] P. Zhang, B. Zhang, D. Chen, L. Yuan, and F. Wen, "Cross-domain correspondence learning for exemplar-based image translation," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[21] T. Grigoryev, A. Voynov, and A. Babenko, "When, why, and which pretrained GANs are useful?" in *Proc. of the International Conference on Learning Representations (ICLR)*, 2022.

[22] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila, "Training generative adversarial networks with limited data," in *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[23] S. Mo, M. Cho, and J. Shin, "Freeze the discriminator: a simple baseline for fine-tuning gans," *arXiv*, 2020.

[24] J. Chen and X. Ran, "Deep learning with edge computing: A review," *Proc. of the IEEE*, 2019.

[25] F. Wang, M. Zhang, X. Wang, X. Ma, and J. Liu, "Deep learning for edge computing applications: A state-of-the-art survey," *IEEE Access*, 2020.

[26] G. Li, X. Ma, X. Wang, L. Liu, J. Xue, and X. Feng, "Fusion-catalyzed pruning for optimizing deep learning on intelligent edge devices," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2020.

[27] X. Sun, J. Choi, C.-Y. Chen, N. Wang, S. Venkataramani, V. V. Srinivasan, X. Cui, W. Zhang, and K. Gopalakrishnan, "Hybrid 8-bit floating point (hfp8) training and inference for deep neural networks," in *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[28] N. Wang, J. Choi, D. Brand, C.-Y. Chen, and K. Gopalakrishnan, "Training deep neural networks with 8-bit floating point numbers," in *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[29] E. Kravchik, F. Yang, P. Kisilev, and Y. Choukroun, "Low-bit quantization of neural networks for efficient inference," in *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, 2019.

[30] Y. Gorbachev, M. Fedorov, I. Slavutin, A. Tugarev, M. Fatekhov, and Y. Tarkan, "Openvino deep learning workbench: Comprehensive analysis and tuning of neural networks inference," in *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, 2019.

[31] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash

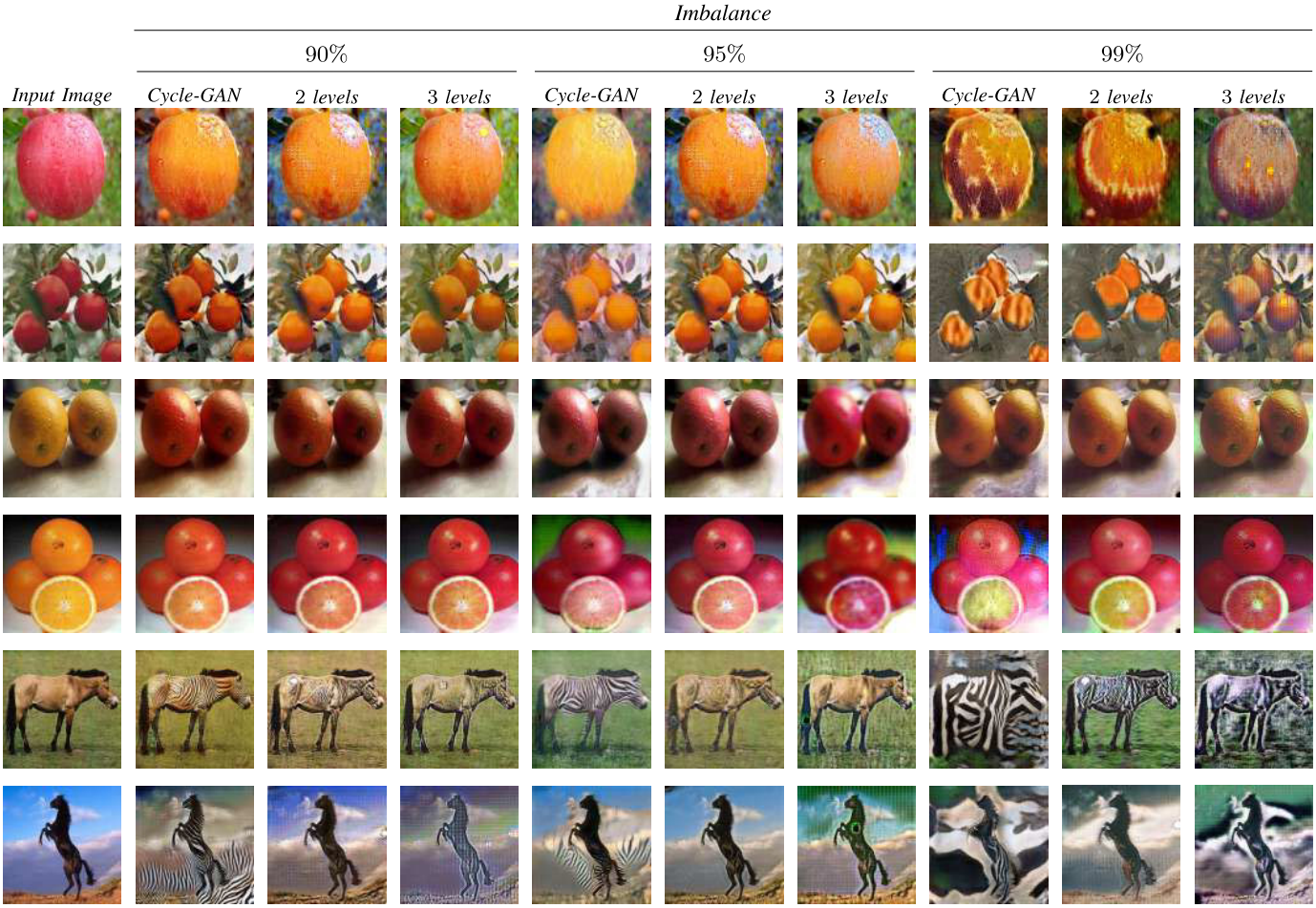


Fig. 2: Qualitative results for different imbalance values, for Apple → Orange, Orange → Apple and Horse → Zebra datasets. Several input images present a more powerful representation of the background compared to the Cycle-GAN baseline. Images from Horse → Zebra appear more challenging increasing the overfitting impact. When $\mathcal{I} = 99\%$ a noticeable degradation is clearly visible.

(a) (b) (c)

Fig. 3: Qualitative comparison for Apple → Orange transformation using the input image (a), and a 2-levels and 3-levels binary tree (shown in (b) and (c), respectively) training from scratch the generators at each level. $\mathcal{I}$ is set to 95%.

(a) (b) (c)

Fig. 4: Qualitative results for Apple → Orange transformation after quantizing and fine-tuning the Cycle-GAN model for 2 (b) and 5 (c) epochs, respectively, using the input images shown in (a).

equilibrium," in *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[32] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.