Palette: Image-to-Image Diffusion Models

Anonymous Author(s)
Affiliation
Address
email

Abstract

We introduce *Palette*, a simple unified framework for image-to-image translation using conditional diffusion models. We demonstrate the effectiveness of Palette on four distinct image-to-image translation tasks, namely: colorization, inpainting, uncropping, and JPEG decompression. We propose a unified evaluation protocol across these tasks on the ImageNet dataset and report several perceptual evaluation metrics including FID, Inception Score, Classification Accuracy of a pre-trained classifier, and Perceptual Distance with the reference image. We outperform existing GAN and autoregressive models on colorization, demonstrating the versatility and the general applicability of diffusion models to image-to-image tasks without the need for any task-specific loss modification, tuning, or architecture modification.

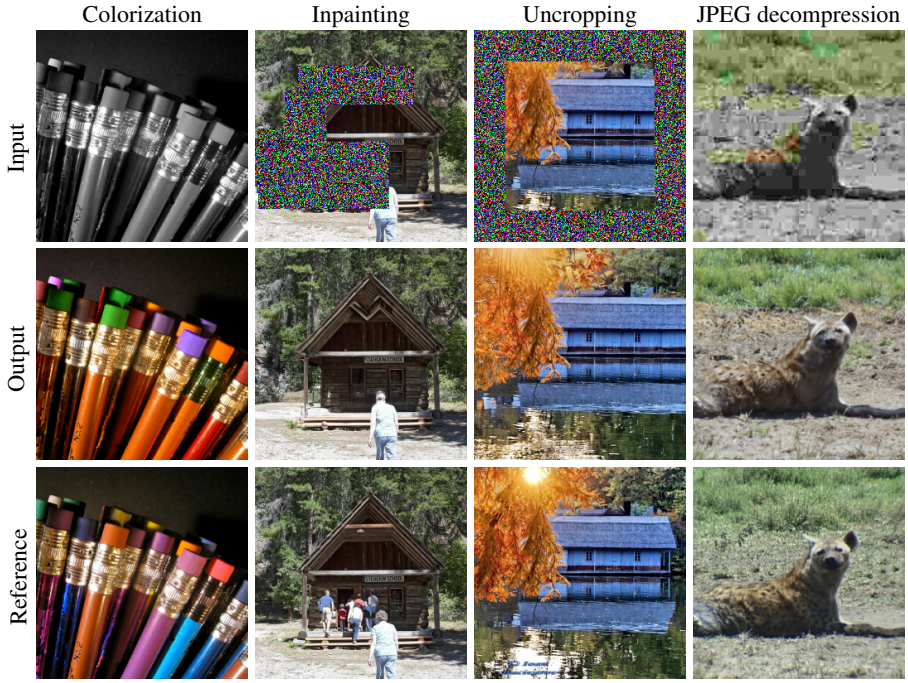

Figure 1: Four examples of Palette's predictions *vs.* the reference images on four different image-to-image translation tasks: colorization, inpainting, uncropping, and JPEG decompression.

Submitted to NeurIPS 2021 Workshop on DGMs and Downstream Applications. Do not distribute.

1 Introduction

Many problems in computer vision and image processing can be formulated as image-to-image translation. Examples include restoration tasks, such as super-resolution, colorization, and inpainting, as well as pixel-level image understanding tasks, such as instance segmentation or the estimation of intrinsic images. Many such tasks, like those in Figure 1, are complex one-to-many inverse problems, where multiple output images may be consistent with a single input. An effective and unified solution to the general problem of image-to-image translation can address many vision problems and enable new creative applications.

Deep generative models present a natural approach to image-to-image translation through learning the conditional distribution of output images given the input image. Generative model families that can capture highly multimodal conditional distributions in the high-dimensional space of images are desirable. Generative Adversarial Networks (GANs) [1] have emerged as the model family of choice for many image-to-image translation tasks [2]; they are capable of generating high fidelity outputs, are broadly applicable, and offer efficient sampling. However, GANs can be challenging to train [e.g., 3, 4], and often drop some modes in the output distribution [e.g., 5, 6]. Other families of generative models such as Auto-regressive Models [7, 8, 9], VAEs [10, 11], and Normalizing Flows [12, 13] have seen success in specific image translation applications, but have not established themselves as a powerful alternative to GANs with the same level of sample quality and generality.

Diffusion models [14, 15, 16] represent a new class of deep generative models built on VAEs and energy-based models, which have received a surge of recent interest [e.g., 17, 18, 19, 20, 21, 22], resulting in key recent results in modeling continuous data distributions. On speech synthesis, diffusion models have achieved on par human evaluation scores with SoTA auoregressive models [23, 24, 25]. On the class-conditional ImageNet generation challenge, they have outperformed the strongest GAN baselines in terms of FID scores [26, 27, 28]. On image super-resolution, they have delivered impressive face enhancement results, outperforming GAN baselines [29].

Despite the recent success of diffusion models, it is not clear whether they can present a robust and unified solution across an array of vision tasks, challenging GAN counterparts on image-to-image translation tasks. This paper investigates the general applicability of image-to-image diffusion models on a suite of four distinct and challenging tasks, namely colorization, inpainting, uncropping, and JPEG decompression (see Figure 1). Our empirical results suggest that the same identical diffusion model, dubbed *Palette*, with no tuning of the architecture details and no changes to the hyper-parameters or the loss function, can deliver high-fidelity outputs across the four aforementioned tasks, outperforming several existing baselines on colorization by a large margin, improving upon... on inpainting, and producing better results on uncropping than ...

In addition, we study various aspects of diffusion models such as the choice of L2 [] vs. L1 [] in the denoising objective. We find that the L2 objective produces a higher degree of diversity between models outputs while both loss functions produce similar sample-quality metrics. Finally, we present a set of well-curated validation protocols for the tasks above based on the ImageNet dataset and report various sample quality metrics to encourage large-scale empirical evaluation of image-to-image translation models on these tasks. We contribute a powerful regression baseline based on a large and well-trained U-Net architecture that serve as a strong baseline for image-to-image translation models.

Our contributions is as follows:

1. We introduce Palette, a general image-to-image translation framework using conditional diffusion models.
2. We apply Palette on four distinct image-to-image translation tasks, namely, colorization, inpainting, uncropping and JPEG decompression.
3. We demonstrate the potential of Palette on ImageNet, a challenging dataset with high diversity.

2 Palette

2.1 Diffusion Models

Palette is a conditional diffusion model [14, 16]. We are given an input-output image pairs $(\boldsymbol{x}, \boldsymbol{y}_0)$, for example in colorization, the input image $\boldsymbol{x}$ would be the gray scale image, while the output

Algorithm 1 Training a denoising model f_θ

1: **repeat**
2: $(\boldsymbol{x}, \boldsymbol{y}_0) \sim p(\boldsymbol{x}, \boldsymbol{y})$
3: $\gamma \sim p(\gamma)$
4: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: Take a gradient descent step on
$\nabla_\theta \left\| f_\theta(\boldsymbol{x}, \sqrt{\gamma}\boldsymbol{y}_0 + \sqrt{1-\gamma}\boldsymbol{\epsilon}, \gamma) - \boldsymbol{\epsilon} \right\|_p^p$
6: **until** converged

Algorithm 2 Inference in T iterative refinement steps

1: $\boldsymbol{y}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \ldots, 1$ **do**
3: $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\boldsymbol{z} = \mathbf{0}$
4: $\boldsymbol{y}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{y}_t - \frac{1-\alpha_t}{\sqrt{1-\gamma_t}} f_\theta(\boldsymbol{x}, \boldsymbol{y}_t, \gamma_t)\right) + \sqrt{1-\alpha_t}\boldsymbol{z}$
5: **end for**
6: **return** $\boldsymbol{y}_0$

image $\boldsymbol{y}_0$ would be the reference colour image. Our goal is to model the image-to-image translation distribution $p(\boldsymbol{y}_0|\boldsymbol{x})$. Palette models this through the use of conditional diffusion models [23, 29].

2.1.1 Forward Process

The Diffusion model is described by a forward and reverse diffusion process. The forward diffusion process is a Markovian process that iteratively adds Gaussian noise to the signal $\boldsymbol{y}_0$ over T iterations:

$$q(\boldsymbol{y}_{t+1}|\boldsymbol{y}_t) = \mathcal{N}(\boldsymbol{y}_{t-1}; \sqrt{\alpha_t}\boldsymbol{y}_{t-1}, (1-\alpha_t)I \tag{1}$$

$$q(\boldsymbol{y}_{1:T}|\boldsymbol{y}_0) = \prod_{t=1}^{T} q(\boldsymbol{y}_t|\boldsymbol{y}_{t-1}) \tag{2}$$

where α_t are hyper-parameters of the noise schedule. The forward process with α_t is constructed in a manner where at $t = T$, $\boldsymbol{y}_T$ is virtually indistinguishable from Gaussian noise. Note, we can also marginalize the forward process at each step:

$$q(\boldsymbol{y}_t|\boldsymbol{y}_0) = \mathcal{N}(\boldsymbol{y}_t; \sqrt{\gamma_t}\boldsymbol{y}_0, (1-\gamma_t)I) \tag{3}$$

where $\gamma_t = \prod_{t'}^{t} \alpha'_t$.

The Gaussian parameterization of the forward process also allows a closed form formulation of the posterior distribution of $\boldsymbol{y}_{t-1}$ given $(\boldsymbol{y}_0, \boldsymbol{y}_t)$ as

$$\begin{aligned} q(\boldsymbol{y}_{t-1} \mid \boldsymbol{y}_0, \boldsymbol{y}_t) &= \mathcal{N}(\boldsymbol{y}_{t-1} \mid \boldsymbol{\mu}, \sigma^2 \boldsymbol{I}) \\ \boldsymbol{\mu} &= \frac{\sqrt{\gamma_{t-1}}\,(1-\alpha_t)}{1-\gamma_t}\,\boldsymbol{y}_0 + \frac{\sqrt{\alpha_t}\,(1-\gamma_{t-1})}{1-\gamma_t}\boldsymbol{y}_t \\ \sigma^2 &= \frac{(1-\gamma_{t-1})(1-\alpha_t)}{1-\gamma_t}\ . \end{aligned} \tag{4}$$

This result proves to be very helpful during inference as shown below.

2.1.2 Learning the Reverse Process

The reverse diffusion process undos the forward process. Palette learns the reverse process. Given a noisy image $\widetilde{\boldsymbol{y}}$,

$$\widetilde{\boldsymbol{y}} = \sqrt{\gamma}\,\boldsymbol{y}_0 + \sqrt{1-\gamma}\,\boldsymbol{\epsilon}\ , \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})\ , \tag{5}$$

the goal is to recover the target image $\boldsymbol{y}_0$. We parameterize our neural network model $f_\theta(x, \widetilde{\boldsymbol{y}}, \gamma)$ to condition on the input x, noisy image $\widetilde{\boldsymbol{y}}$ and the current noise level γ. The objective function is predict the noise vector $\boldsymbol{\epsilon}$:

$$\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})}\mathbb{E}_{\boldsymbol{\epsilon},\gamma}\left\| f_\theta(\boldsymbol{x}, \underbrace{\sqrt{\gamma}\,\boldsymbol{y}_0 + \sqrt{1-\gamma}\,\boldsymbol{\epsilon}}_{\widetilde{\boldsymbol{y}}}, \gamma) - \boldsymbol{\epsilon} \right\|_p^p\ , \tag{6}$$

This is also known as L_{simple} in [16]. This is objective is also equivalent to maximizing a weighted variational lower-bound on the likelihood [16].

2.1.3 Inference

Palette performs inference via the learned reverse process. Recall that the forward process is constructed in such a manner that the prior distribution p(y_T) is approximately distributed according to standard normal distribution $\mathcal{N}(\boldsymbol{y}_T|\mathbf{0}, \boldsymbol{I})$. This allows the sampling process to start at pure Gaussian noise, followed by T steps of iterative refinement.

Recall that our neural network model f_θ is trained to estimate $\boldsymbol{\epsilon}$, given any noisy image $\widetilde{\boldsymbol{y}}$ including $\boldsymbol{y}_t$. Thus, given $\boldsymbol{y}_t$, we approximate $\boldsymbol{y}_0$ by rearranging the terms in (5) as

$$\hat{\boldsymbol{y}}_0 = \frac{1}{\sqrt{\gamma_t}} \left(\boldsymbol{y}_t - \sqrt{1-\gamma_t}\, f_\theta(\boldsymbol{x}, \boldsymbol{y}_t, \gamma_t) \right) \ . \tag{7}$$

Following the formulation of [16], we substitute our estimate $\hat{\boldsymbol{y}}_0$ into the posterior distribution of $q(\boldsymbol{y}_{t-1}|\boldsymbol{y}_0, \boldsymbol{y}_t)$ in (4) to parameterize the mean of $p_\theta(\boldsymbol{y}_{t-1}|\boldsymbol{y}_t, \boldsymbol{x})$ as

$$\mu_\theta(\boldsymbol{x}, \boldsymbol{y}_t, \gamma_t) = \frac{1}{\sqrt{\alpha_t}} \left(\boldsymbol{y}_t - \frac{1-\alpha_t}{\sqrt{1-\gamma_t}} f_\theta(\boldsymbol{x}, \boldsymbol{y}_t, \gamma_t) \right) \ , \tag{8}$$

and we set the variance of $p_\theta(\boldsymbol{y}_{t-1}|\boldsymbol{y}_t, \boldsymbol{x})$ to $(1-\alpha_t)$, a default given by the variance of the forward process [16].

Following this parameterization, each iteration of the reverse process can be demonstrated as,

$$\boldsymbol{y}_{t-1} \leftarrow \frac{1}{\sqrt{\alpha_t}} \left(\boldsymbol{y}_t - \frac{1-\alpha_t}{\sqrt{1-\gamma_t}} f_\theta(\boldsymbol{x}, \boldsymbol{y}_t, \gamma_t) \right) + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_t \ ,$$

where $\boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$. This resembles one step of Langevin dynamics with f_θ providing an estimate of the gradient of the data log-density.

2.2 Architecture

Palette uses the standard U-Net architecture from [16]. We incorporate the following modifications to the architecture inspired by recent publications [18, 29, 28]:

- Using BigGAN [30] residual blocks.
- Increasing the number of self-attention heads.
- Using Adaptive Group Normalization [28].

Following [29], we condition Palette on the source image by concatenating it with the noisy image input to the U-Net model (Figure 2).

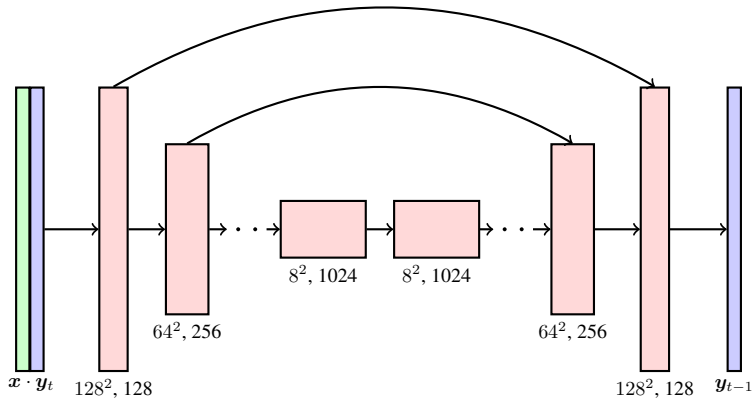

Figure 2: U-Net architecture from [29]. The source image $\boldsymbol{x}$ is simply concatenated to the noisy image $\boldsymbol{y}_t$ to add source image conditioning to the reverse process.

3 Experiments

3.1 Training Details

Palette's neural architecture is based off the 256×256 class-conditional ablated U-Net model used by [28]. The two main differences between our architecture and theirs are (i) absence of class-conditioning, and (ii) additional conditioning of the source image through concatenation. The conditioning of the source image through concatenation is similar to [29]. It is important to note that we are primarily interested in pushing the limits of these diffusion models for image-to-image translation tasks, hence we primarily train models with order of magnitude more parameters than the ones previously used in the existing literature. We use a batch size of 1024, and train each model for 1M training steps. We do not find overfitting to be an issue in this setting.

3.2 Evaluation Protocol

Evaluation on image-to-image translation tasks is a challenging problem, and existing works adopt different evaluation protocols for different image-to-image translation tasks. For example, in colorization, prior work [31, 7, 32] evaluate using FID score and human evaluation, while tasks like inpainting [33, 34], JPEG decompression [35, 36, 37], and extrapolation [38, 39] primarily rely on qualitative comparisons, and pixel wise comparison metrics such as PSNR/SSIM. In this work, we analyze the effectiveness and limitations of these commonly used evaluation metrics for each of these translation tasks.

Automated Metrics : For each task, we report a number of perceptual scores including:

1. Inception Score (IS)
2. Fréchet Inception Distance (FID)
3. Classification Accuracy (CA) : Top-1% accuracy of a pre-trained ResNet-50 classifier.
4. Perceptual Distance (PD) : Euclidean distance in the Inception-v1 feature space.

3.3 Tasks

In order to demonstrate the effectiveness of diffusion models for conditional image generation, we show results on a variety of image-to-image translation tasks - (i) Colorization, (ii) Inpainting, (iii) Extrapolation, (iv) JPEG Decompression.

3.3.1 Colorization

For colorization, we use the grayscale image as the source image, and train the model to generate the RGB image. Inspired by existing works on colorization [32, 31], we experimented with different colorspace parameterizations such as YCbCr and LAB colorspace, but found them to yield similar results. We chose RGB parameterization in order to maintain generality among different translation tasks.

Evaluation Protocol : Following existing works [31, 7, 32], we train and evaluate our models on ImageNet [40] dataset. Following ColTran [32], we use 1st 5000 images from ImageNet validation set to report performance on standard metrics and use next 5000 validation images as the reference distribution for FID.

Results : Palette outperforms existing SoTA ColTran [32] by a significant margin in terms of FID, bridging the gap with the reference images. Interestingly, our simple L2 Regression baseline is also able to outperform the existing comparably sophisticated methods, indicating the effectiveness of scaling the model capacity.

Sample Diversity : Generating diverse set of colorized images for a given gray-scale input is one of the primary challenges in image colorization task. Here, we analyze the sample diversity of diffusion models for colorization. Specifically, we analyze the impact of the norm of the diffusion objective function L_{simple} [16] on sample diversity. Inspired by [7], we generate multiple samples for an input grayscale image, and compute pairwise multi-scale SSIM between of 1st generated image with the rest. We repeat this process for multiple grayscale images, and plot the distribution of SSIM values

Model	FID-5K ↓	IS ↑	CA ↑	PD ↓
Prior Work				
cGAN [41]	24.41	-	-	-
PixColor [7]	24.32	-	-	-
Coltran [32]	19.37	-	-	-
Our Work				
Regression	17.89	169.8	68.2%	60.0
Palette	**15.78**	**200.8**	**72.5%**	**46.2**
Ground Truth	14.68	229.6	75.6%	0.0

Table 1: Fréchet Inception Distance (FID)-5k, Inception Score (IS), Classification Accuracy Score (CAS), and Perceptual Distance (PD) on ImageNet Colorization.

Figure 3: Comparison of different methods for colorization on ImageNet validation images.

in Figure 4. Lesser diversity implies higher similarity among sample generations resulting in right shifted SSIM distribution. We observe a significant such right shift for our model trained with L1 norm compared to the models trained with L2 norm, indicating that training with L2 norm promotes significantly more sample diversity.

3.3.2 Inpainting

Given an input image with a masked region, the goal of inpainting is to fill the mask in order to produce realistic sample generations. One of the key components that defines the scope of inpainting task is the distribution of masks. Following recent work by [33], we train our models on free-form generated masks. We also augment these with simpler rectangular masks with varying aspect ratios. In order to maintain generality among the tasks, we do not pass an additional binary mask channel unlike existing methods and simply fill the masked region with standard gaussian noise. During training, we only train the model to predict the masked spatial region instead of the entire image. In our initial experiments, we found this modification to help in faster convergence. During sample generation, we only use the model prediction for the masked spatial region.

Evaluation Protocol : Consistent with previous works [34], we use Places365-Challenge [42] dataset for training. We additionally augment the training dataset with ImageNet in order to assess the performance of our model on this more challenging dataset. We perform evaluation on ImageNet and Places2 validation set. For ImageNet, we use the class-balanced 10k image subset ctest10k [43]. For Places2, we construct a similar category-balanced subset of 10,950 images (30 images per category). We call this subset places10k. Based on the area covered by the masked region, we report results for two categories - i) 10-20% region, and ii) 20-30% region.

Results : Table 2 shows the performance of Palette for different inpainting configurations and datasets. Palette exhibits strong performance outperforming DeepFillv2 [33] on all benchmarks. It is important

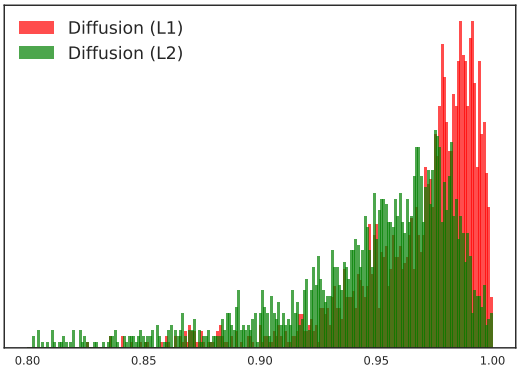

Figure 4: Distribution of pairwise multi-scale SSIM: Quantitative comparison of sample diversity between diffusion models trained with L1 vs L2 norm of L_{simple}. For each image, we generate 16 samples, and compute the pairwise multi-scale SSIM for the first generation with rest of the 15 generations. Higher SSIM implies higher similarity among the multiple samples, hence less diversity.

to note that comparison with DeepFillv2 on ImageNet is unfair, as it is not trained on ImageNet dataset.

Model	**ImageNet**				**Places365**	
	FID $\downarrow$	**IS** $\uparrow$	**CA** $\uparrow$	**PD** $\downarrow$	**FID** $\downarrow$	**PD** $\downarrow$
10-20% Mask						
DeepFillv2 [33]	6.7	198.2	71.6%	38.6	12.2	38.1
Palette (Ours)	**5.2**	**219.2**	**73.7%**	**15.5**	**11.5**	**20.5**
20-30% Mask						
DeepFillv2 [33]	9.4	174.6	68.8%	64.7	13.5	63.0
Palette (Ours)	**5.2**	**208.3**	**72.2%**	**27.4**	**11.6**	**34.5**
Reference	5.1	231.6	74.6%	0.0	11.4	0.0

Table 2: Quantitative evaluation on free-form Inpainting on ImageNet and Places365. The images are center cropped to 256×256 resolution.

3.3.3 Uncropping

The task of image uncropping or extrapolation involves extending a given image along one or more sides. Uncropping is a relatively loosely defined task. While some existing works [38, 44] focus on extending varying portion of image only along a specific side (sometimes also called image extension), other works have more flexible setups [45, 39]. We train Palette for uncropping uniformly along all four sides as well as only a particular side from any direction. We fix the uncropping region to 50% of image area for both settings. Similar to inpainting, we mask the training loss to only the masked region, and use the model prediction only from the masked region during generation.

Evaluation Protocol: Many existing methods [38, 45] train their uncropping model on varying subsets of the Places365-Challenge dataset. We keep our setup similar to inpainting, and train on both ImageNet and Places365-Challenge dataset. While Palette can handle uncropping uniformly from all 4 sides as well as any arbitrary side, in order to fairly compare against existing methods, we evaluate Palette for specifically for uncropping the right half of the image. For ImageNet, we report results on the ctest10k subset. Since [38] is only trained on the 50 categories of Places2, we specifically evaluate on the validation images from these categories to ensure fair comparison.

Results: Table 3 summarizes the performance of Palette on both datasets. Palette outperforms [38] on both datasets by a significant margin demonstrating the strong capability of diffusion models to perform image uncropping without any task specific engineering. Figure 5 shows a qualitative comparison between Palette and [38].

Model	ImageNet				Places365	
	FID ↓	IS ↑	CA ↑	PD ↓	FID ↓	PD ↓
Prior Work						
Boundless [38]	18.7	104.1	58.8%	127.9	28.3	115.0
Our Work						
Palette (Ours)	**5.8**	**138.1**	**63.4%**	**85.9**	**22.9**	**93.4**
Reference	2.9	254.4	76.5%	0.0	23.6	0.0

Table 3: Quantitative evaluation on uncropping.

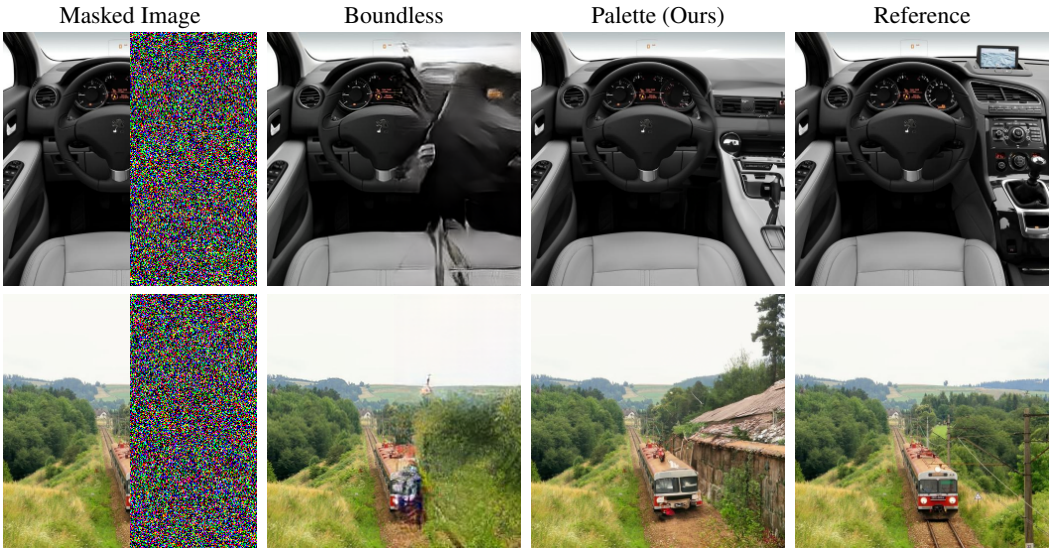

Figure 5: Comparison of different methods for Image Uncropping on Places365 validation images.

3.4 JPEG Decompression

We further demonstrate the performance of Palette on the task of JPEG artifact removal - one of the long standing image enhancement problems [35, 37, 36]. Similar to existing works [46, 36], we train Palette to perform JPEG artifact removal at simultaneously at multiple quality factors. While previous works are generally limited upto a Quality Factor of 10, we go a step further and train Palette to remove artifacts of upto a Quality Factor of 5. During training, we use an exponential distribution to sample the quality factor, ensuring significantly more training on lower quality factors.

Evaluation Protocol: In order to be consistent with other tasks, and for benchmarking purposes, we perform training and evaluation on ImageNet dataset. Note that this is unlike existing works [35, 36] which mainly use small datasets such as DIV2K / BSD500 for training and evaluation. Recent works such as [37] use a relatively relatively larger MS-COCO dataset for training, however, to the best of our knowledge, we are the first to train and evaluate JPEG artifact removal on ImageNet. We compare Palette with a strong Regression baseline which uses an identical architecture. We report results on quality factors of 5, 10 and 20.

Results: Table 4 shows the performance of Palette on ImageNet. Palette exhibits strong performance across all quality factors, being significantly better than the Regression baseline. The performance gap between Palette and Regression baseline widens with decreasing quality factor. Figure 6 shows the qualitative comparison between Palette and Regression baseline at quality factor of 5. It is easy to see that the regression model produces blurry looking outputs, while Palette produces sharp images.

Figure 6: Comparison of different methods for JPEG decompression on ImageNet validation images.

Model	**FID-5K** ↓	**IS** ↑	**CA** ↑	**PD** ↓
Quality Factor = 20				
Regression	11.5	158.7	69.7%	65.4
Palette (Ours)	**4.3**	**208.7**	**73.5%**	**37.1**
Quality Factor = 10				
Regression	18.0	117.2	63.5%	102.2
Palette (Ours)	**5.4**	**180.5**	**70.7%**	**58.3**
Quality Factor = 5				
Regression	29.0	73.9	52.8%	155.4
Palette (Ours)	**8.3**	**133.6**	**64.2%**	**95.5**
Ground Truth	2.7	250.1	75.9%	0.0

Table 4: Quantitative evaluation on JPEG Decompression.

4 Conclusion

We presented Palette, a unified image-to-image translation diffusion model. We demonstrate the performance of Palette on 4 distinct image-to-image translation tasks. These results showcase the versatility of diffusion models for conditional image generation without the need for any task specific modifications unlike other generative models such as GANs.

References

[1] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Networks. *NIPS*, 2014.

[2] Phillip Isola, Jun-Yan Zhu, and Tinghui Zhou ajnd Alexei A. Efros. Image-to-Image Translation with Conditional Adversarial Nets. In *CVPR*, 2017.

[3] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein GAN. In *arXiv*, 2017.

[4] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans. *arXiv preprint arXiv:1704.00028*, 2017.

[5] Luke Metz, Ben Poole, David Pfau, and Jascha Sohl-Dickstein. Unrolled generative adversarial networks. *arXiv preprint arXiv:1611.02163*, 2016.

[6] Suman Ravuri and Oriol Vinyals. Classification accuracy score for conditional generative models. *arXiv preprint arXiv:1905.10887*, 2019.

[7] Sergio Guadarrama, Ryan Dahl, David Bieber, Mohammad Norouzi, Jonathon Shlens, and Kevin Murphy. Pixcolor: Pixel recursive colorization. *arXiv preprint arXiv:1705.07208*, 2017.

[8] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *ICML*, 2018.

[9] Aaron van den Oord, Nal Kalchbrenner, Oriol Vinyals, Lasse Espeholt, Alex Graves, and Koray Kavukcuoglu. Conditional image generation with PixelCNN decoders. In *NIPS*, pages 4790–4798, 2016.

[10] Diederik P Kingma and Max Welling. Auto-Encoding Variational Bayes. In *ICLR*, 2013.

[11] Arash Vahdat and Jan Kautz. NVAE: A deep hierarchical variational autoencoder. In *NeurIPS*, 2020.

[12] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. *arXiv:1605.08803*, 2016.

[13] Diederik P. Kingma and Prafulla Dhariwal. Glow: Generative Flow with Invertible 1x1 Convolutions. In *NIPS*, 2018.

[14] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, pages 2256–2265. PMLR, 2015.

[15] Yang Song and Stefano Ermon. Improved Techniques for Training Score-Based Generative Models. *arXiv preprint arXiv:2006.09011*, 2020.

[16] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv preprint arXiv:2006.11239*, 2020.

[17] Ruojin Cai, Guandao Yang, Hadar Averbuch-Elor, Zekun Hao, Serge Belongie, Noah Snavely, and Bharath Hariharan. Learning Gradient Fields for Shape Generation. In *ECCV*, 2020.

[18] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-Based Generative Modeling through Stochastic Differential Equations. In *ICLR*, 2021.

[19] Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Towards non-autoregressive language models. *arXiv preprint arXiv:2102.05379*, 2021.

[20] Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. *arXiv preprint arXiv:2106.05931*, 2021.

[21] Diederik P Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *arXiv preprint arXiv:2107.00630*, 2021.

[22] Jacob Austin, Daniel Johnson, Jonathan Ho, Danny Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. *arXiv preprint arXiv:2107.03006*, 2021.

[23] Nanxin Chen, Yu Zhang, Heiga Zen, Ron J. Weiss, Mohammad Norouzi, and William Chan. WaveGrad: Estimating Gradients for Waveform Generation. In *ICLR*, 2021.

[24] Nanxin Chen, Yu Zhang, Heiga Zen, Ron J. Weiss, Mohammad Norouzi, Najim Dehak, and William Chan. WaveGrad 2: Iterative Refinement for Text-to-Speech Synthesis . In *INTERSPEECH*, 2021.

[25] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. DiffWave: A Versatile Diffusion Model for Audio Synthesis. *ICLR*, 2021.

[26] Alex Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. *arXiv preprint arXiv:2102.09672*, 2021.

[27] Jonathan Ho, Chitwan Saharia, William Chan, David J. Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded Diffusion Models for High Fidelity Image Generation. In *arXiv*, 2021.

[28] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. *arXiv preprint arXiv:2105.05233*, 2021.

[29] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J. Fleet, and Mohammad Norouzi. Image Super-Resolution via Iterative Refinement. *arXiv preprint arXiv:2104.07636*, 2021.

[30] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[31] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016.

[32] Manoj Kumar, Dirk Weissenborn, and Nal Kalchbrenner. Colorization transformer. In *ICLR 2021*, 2021.

[33] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4471–4480, 2019.

[34] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5505–5514, 2018.

[35] Chao Dong, Yubin Deng, Chen Change Loy, and Xiaoou Tang. Compression artifacts reduction by a deep convolutional network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 576–584, 2015.

[36] Pengju Liu, Hongzhi Zhang, Kai Zhang, Liang Lin, and Wangmeng Zuo. Multi-level wavelet-cnn for image restoration. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 773–782, 2018.

[37] Leonardo Galteri, Lorenzo Seidenari, Marco Bertini, and Alberto Del Bimbo. Deep universal generative adversarial compression artifact removal. *IEEE Transactions on Multimedia*, 21(8):2131–2145, 2019.

[38] Piotr Teterwak, Aaron Sarna, Dilip Krishnan, Aaron Maschinot, David Belanger, Ce Liu, and William T Freeman. Boundless: Generative adversarial networks for image extension. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10521–10530, 2019.

[39] Yi Wang, Xin Tao, Xiaoyong Shen, and Jiaya Jia. Wide-context semantic image extrapolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1399–1408, 2019.

[40] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[41] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[42] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[43] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Learning representations for automatic colorization. In *European conference on computer vision*, pages 577–593. Springer, 2016.

[44] Richard Strong Bowen, Huiwen Chang, Charles Herrmann, Piotr Teterwak, Ce Liu, and Ramin Zabih. Oconet: Image extrapolation by object completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2307–2317, June 2021.

[45] Yen-Chi Cheng, Chieh Hubert Lin, Hsin-Ying Lee, Jian Ren, Sergey Tulyakov, and Ming-Hsuan Yang. In&out: Diverse image outpainting via gan inversion. *arXiv preprint arXiv:2104.00675*, 2021.

[46] Max Ehrlich, Larry Davis, Ser-Nam Lim, and Abhinav Shrivastava. Quantization guided jpeg artifact correction. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VIII 16*, pages 293–309. Springer, 2020.