Towards Fast Generative Image Compression

Nathan Dalal
Stanford University
Stanford, CA 94305
nathanhd@stanford.edu

Nikhil Sardana
Stanford University
Stanford, CA 94305
nsardana@stanford.edu

Zhilin Jiang
Stanford University
Stanford, CA 94305
zjiang23@stanford.edu

1 Introduction

Figure 1: From left to right: the original image (9.5 megabytes), the BPG compressed version (13.2 kilobytes), and the JPG compressed version (13.3 kilobytes). Crops correspond to the image directly above them. Image credit: Seguin [2010].

Image compression using deep neural networks has been a focus of recent research. Santurkar et al. [2017], Rippel and Bourdev [2017], and Agustsson et al. [2018] have explored using generative adversarial networks for image compression, and have demonstrated quality comparable to or better than standard compression techniques (JPEG, JPEG-2000, and BPG), using lower bitrates.

However, previous research primarily focused on compression quality over speed. Since generative models require significantly more computation than standard compression techniques, the applications of generative compression are severely limited. In addition, extreme compression to less than 0.1 bits per pixel, which was explored in Agustsson et al. [2018], is most useful in situations with extremely limited bandwidth. For example, extreme compression may be useful for video and image messaging in developing countries with nascent infrastructure. In these cases, devices are unlikely to have the computation power necessary to decode images in a reasonable time. In this paper, we work towards reducing inference (decompression) time of generative compression, bringing it closer to real-world usability. Since image quality is often correlated with model size, it is challenging to reduce parameters and decrease inference time while maintaining output quality.

Preprint. Work in progress.

2 Related Work

Santurkar et al. [2017] was the first to use a generative adversarial network [Goodfellow et al., 2014] for image compression. First, they pre-trained a decoder network adversarially with a discriminator, using the standard GAN loss function. Then, they trained an encoder to minimize the the pixel-level and perceptual losses.

Rippel and Bourdev [2017] took a slightly different approach, modifying the traditional GAN discriminator to take in two input images and output which of the two images was generated. In addition, outputs are taken at various layers throughout the discriminator, rather than only at the final layer, to aggregate information from multiple scales.

Agustsson et al. [2018] is the current state of the art in generative compression quality. They first run images through an encoder to generate a compressed representation, and then train a generator on these compressed representations to recover the original image and mimic the data distribution. Finally, the generated representations are fed into a discriminator and the entire network is trained adversarially. The authors report higher image quality than BPG, even when BPG uses 21-49% more bits. The model also compares favorably to other generative compression techniques. Rippel and Bourdev [2017] uses 29–197% more bits than Agustsson et al. [2018], and produces images of inferior quality.

2.1 Compression Speed

Only Rippel and Bourdev [2017] have noted the speed of their compression model, stating that their model encodes an image in 9ms and decodes in 10ms, compared to 18ms/12ms for JPEG and 350ms/80ms for JPEG 2000. However, this comparison is not fair, since the model was run on an Nvidia 980 Ti GPU, while the times for JPEG and JPEG 2000 are from an unnamed CPU.

Santurkar et al. [2017] and Agustsson et al. [2018] make no mention of encoding or decoding time. However, Agustsson et al. [2018] leverages the architecture of the pix2pixHD model [Wang et al., 2017], for its encoder and decoder (generator). The pix2pixHD model takes 20–30 milliseconds for inference on an NVIDIA 1080 Ti with 11GB of VRAM, so the Agustsson et al. [2018] is at least 2–3 times slower than Rippel and Bourdev [2017], even on more powerful hardware. Our work focuses on retaining the quality of the Agustsson et al. [2018] model, while bringing encoding and decoding time under 10ms, in line with Rippel and Bourdev [2017].

3 Task Definition

Our main goal is to reduce the inference (image decompression) time of the model in Agustsson et al. [2018] from greater than 30 milliseconds down to 10 milliseconds without a noticeable decline in quality. In order to reduce the inference time, we must reduce the number of parameters in the generator. Since the generator in Agustsson et al. [2018] is based on the generator in pix2pixHD [Wang et al., 2017], we must therefore reduce the number of parameters in the pix2pixHD generator.

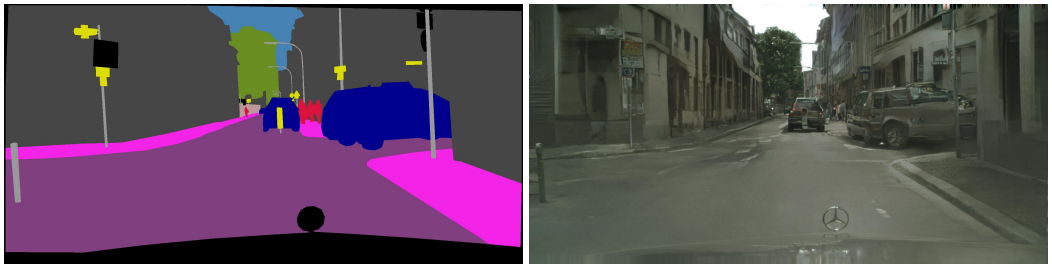


Figure 2: Input semantic label map and corresponding pix2pixHD output.

Pix2pixHD is an image-to-image translation model based on conditional GANs [Mirza and Osindero, 2014]. The generator maps semantic label maps to realistic image scenes, and the discriminator attempts to distinguish between generated scenes and real images. In a conditional GAN, the generator and discriminator operate in a supervised setting. In this case, the training dataset consists of data-label pairs, $\{(\mathbf{s_i}, \mathbf{x_i})\}$, of semantic label maps ($\mathbf{s_i}$) and their corresponding images ($\mathbf{x_i}$).

We modify the pix2pixHD generator to reduce the number of parameters. We train and evaluate our modified pix2pixHD model on the Cityscapes dataset [Cordts et al., 2016], a standard benchmark dataset for segmentation and object detection tasks. This allows for direct speed and quality comparison with the standard pix2pixHD model.

4 Approach

Since Agustsson et al. [2018] is the current state of the art in generative compression quality, we build off their model. Agustsson et al. [2018] leverages the architecture of the pix2pixHD model [Wang et al., 2017], which uses a residual network structure proposed by He et al. [2016]. In pix2pixHD, a small residual network G_1 (the *global* generator) is trained on lower resolution image, and then attached to a larger residual network G_2 (the *local* generator) that includes downsampling and upsampling layers. The entire generator, G, is then trained as a whole using high resolution images. Since Agustsson et al. [2018] used only the global generator from pix2pixHD, we concentrate our modifications to G_1.

To reduce the number of parameters in the pix2pixHD generator, we propose the following modifications:

1. Further downsample the input before the residual blocks in G_1, the global generator.
2. Use Dense blocks [Huang et al., 2017] instead of Residual blocks [He et al., 2016].
3. Reduce the number of blocks in the generators.

DenseNet [Huang et al., 2017] introduced Dense blocks, which build upon the residual connections introduced in He et al. [2016] by directly connecting every layer to every other layer. This modification allowed [Huang et al., 2017] to attain comparable results to He et al. [2016] on CIFAR with far fewer parameters. Intuitively, substituting Residual blocks with Dense blocks should result in similar accuracy improvements for our generator with fewer parameters.

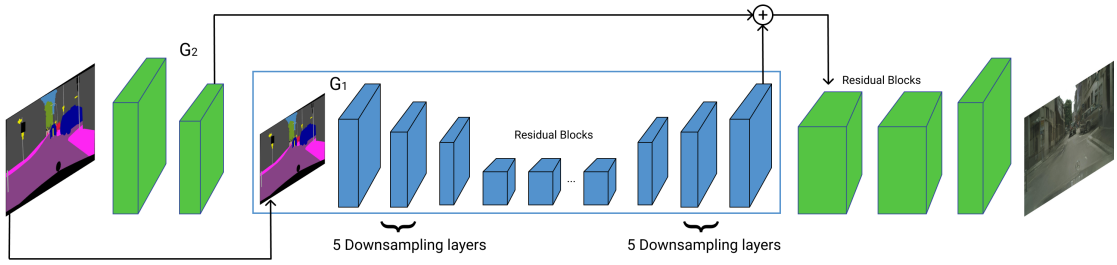


Figure 3: Proposed modification: Extra downsampling layer before residual blocks in global generator.

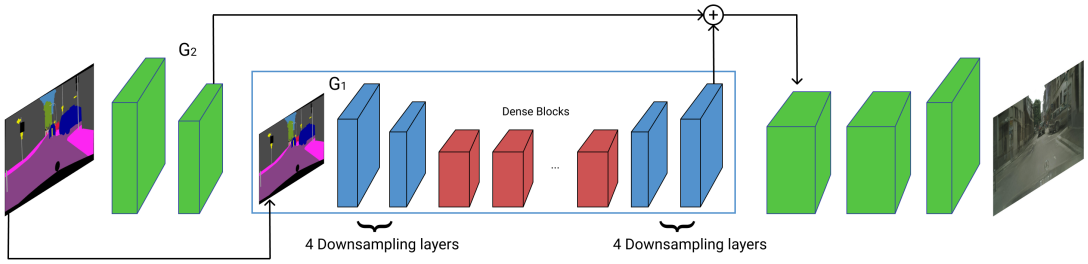


Figure 4: Proposed modification: Replacing residual blocks with Dense Bottleneck blocks.

4.1 Loss Functions

The loss function for our modified pix2pixHD model is identical to the original pix2pixHD loss function in Wang et al. [2017]. Given the generator, G, and multiscale discriminator D_1, D_2, and D_3, we calculate the **feature matching loss** for some semantic label map $\mathbf{s}$ and input image $\mathbf{x}$ using the following equation:

$$\mathcal{L}_{\mathrm{FM}}\left(G, D_k\right)=\mathbb{E}_{(\mathbf{s}, \mathbf{x})} \sum_{i=1}^{T} \frac{1}{N_i}\left[\left\|D_k^{(i)}(\mathbf{s}, \mathbf{x})-D_k^{(i)}(\mathbf{s}, G(\mathbf{s}))\right\|_1\right]$$

where $k = 1, 2, 3$, and $D_k^{(i)}$ represents the ith layer of the k-th scale discriminator. T is the total number of layers, and N_i is the number of elements in the ith layer. The full loss function for Wang et al. [2017] combined the feature matching loss and the traditional GAN loss, summing the two with a scaling factor, λ:

$$\mathcal{L}=\min _G\left(\left(\max _{D_1, D_2, D_3} \sum_{k=1,2,3} \mathcal{L}_{\mathrm{GAN}}\left(G, D_k\right)\right)+\lambda \sum_{k=1,2,3} \mathcal{L}_{\mathrm{FM}}\left(G, D_k\right)\right)$$

$\mathcal{L}_{GAN}$ follows from the standard conditional GAN loss:

$$\mathcal{L}_{GAN}(G, D)=\mathbb{E}_{(\mathbf{s}, \mathbf{x})}[\log D(\mathbf{s}, \mathbf{x})]+\mathbb{E}_{\mathbf{s}}[\log (1-D(\mathbf{s}, G(\mathbf{s}))]$$

4.2 Experiments

We implement the following modifications to the global generator (G_1) of the pix2pixHD model:

Adding one extra pair of downsampling/upsampling layers. Since residual blocks have numerous parameters, reducing the height and width of the ResNet blocks in the generator should reduce parameters. We further downsample the input to the generator G_1 before the residual blocks, and add an extra upsampling layer after the residual blocks. We call this model Extra-Layer-pix2pixHD.

Replacing ResNet blocks with DenseNet. Instead of using 9 residual blocks in the generator, we use 9 Bottleneck Dense Blocks in the model's generator. We call this model Dense-9-pix2pixHD.

Varying the number of Dense blocks. Since Wang et al. [2017] gives no intuition behind using 9 residual blocks, we train Dense-pix2pixHD with 6, 9, and 15 Bottleneck Dense blocks. We call these models Dense-6-pix2pixHD, Dense-9-pix2pixHD, and Dense-15-pix2pixHD.

Varying the growth rate. The growth rate controls the depth of the output of each Dense block. We test growth rates of $k = 4, 32$ and 128. Since the depth of each Dense block output scales linearly with the growth rate, models with larger growth rates should see improvements in image quality, at the expense of increased inference time due to extra parameters.

Adding Spectral Normalization to the Discriminator. Spectral Normalization [Miyato et al., 2018] is a technique used to stabilize the discriminator during GAN training. We experiment with adding Spectral Normalization to the Dense-pix2pixHD discriminator to reduce mode collapse. When implementing Spectral Normalization, we remove the batch normalization present in the standard pix2pixHD discriminator. We call this model Dense-SpectNorm-pix2pixHD.

5 Results

We train all models on the training portion of the Cityscapes dataset [Cordts et al., 2016]. The optimizer, learning rate schedule, weight initialization, instance features, and other hyperparameters remain consistent with Wang et al. [2017]. We compare image quality and inference time with the baseline pix2pixHD model, trained for 60 epochs. All inference times are measured on a Google Cloud Platform virtual machine with an NVIDIA Tesla V100 GPU and 2 vCPUs.

We provide sample outputs from each trained model below.

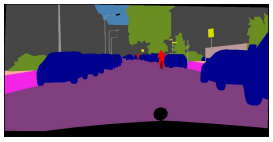
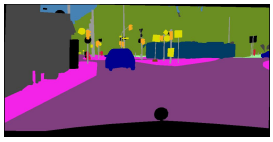
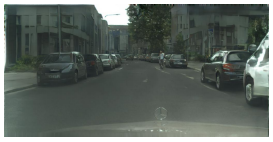
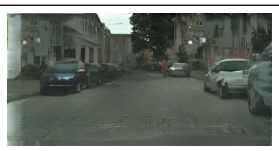
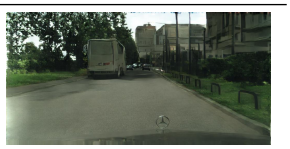
Model	Sample #1	Sample #2	Sample #3
Input Semantic Label			
pix2pixHD Baseline (60 epochs)			
Extra-Layer (60 epochs)			
DenseNet #Blocks = 9 GrowthRate = 32 (60 epochs)			
DenseNet #Blocks = 9 GrowthRate = **4** (60 epochs)			
DenseNet #Blocks = 9 GrowthRate = **128** (60 epochs)			
DenseNet #Blocks = **6** GrowthRate = 32 (60 epochs)			
DenseNet #Blocks = **18** GrowthRate = 32 (60 epochs)			
DenseNet #Blocks = 9 GrowthRate = 32 **w/ SpectralNorm** (60 epochs)			

Table 1: Inputs and generated samples of 3 different test images.

5.1 Metrics

We evaluate both inference speed and image quality of our modified pix2pixHD models. We report the average inference time over fifty test samples for each of our models. We evaluate image quality using both quantitative and qualitative metrics. For our quantitative metric, we report mean Intersection over Union. Qualitatively, we run a user study comparing synthesized images from different techniques.

Model Type	Inference (sec/image)	Difference from Baseline (%)
pix2pixHD (baseline)	0.09475	0
Extra-Layer-pix2pixHD	**0.09229**	**-2.59**
Dense-9-growth-32-pix2pixHD	0.09352	-1.30
Dense-9-growth-4-pix2pixHD	0.09368	-1.13
Dense-9-growth-128-pix2pixHD	0.09361	-1.20
Dense-6-growth-32-pix2pixHD	0.09245	-2.43
Dense-18-growth-32-pix2pixHD	0.09484	+0.10
Dense-9-growth-32-spectNorm-pix2pixHD	**0.09229**	**-2.59**

Table 2: Average inference time over 50 test samples.

Model Type	Human Preference over pix2pixHD
Dense-9-growth-32-pix2pixHD	27.0%
Dense-18-growth-32-pix2pixHD	19.0%

Table 3: Percentage of humans preferring our results to the baseline model. Chance is at 50%.

We calculate mean Intersection over Union (IoU) according to the procedure in Wang et al. [2017]. First, we generate fifty samples from the Cityscapes test set from each model. Then, we run a standard semantic segmentation model, PSPNet [Zhao et al., 2016], on the generated outputs. The PSPNet outputs a predicted semantic segmentation map. Intuitively, the more realistic output our model generates, the closer the PSPNet output semantic label map should be to the ground truth Cityscapes label map. We calculate the mean Intersection over Union between the PSPNet output semantic label map and the Cityscapes ground truth. After outputting the results from PSPNet, we use scripts supplied by Cityscapes [Marius Cordts, 2018] to account for IoU bias towards object classes of larger sizes.

Since mean Intersection over Union is not an accurate representation of human perception, we conduct a human subjective study, similar to Wang et al. [2017]. For the perception study, we take ten semantic label maps and generate output images from three models: the baseline pix2pixHD, Dense-9-growth-32-pix2pixHD, and Dense-18-growth-32-pix2pixHD. Participants are given a form with ten generated output images from the baseline pix2pixHD, and either ten images from Dense-9-growth-32-pix2pixHD or Dense-18-growth-32-pix2pixHD. Users are asked to select which image looks more natural. Images are presented to users unlabeled and the position of the baseline (left vs. right) is randomized to minimize bias. Due to budget constraints, we do not hire paid workers for evaluation, and instead recruit 20 volunteers to compare pix2pixHD samples to Dense-9-growth-32-pix2pixHD samples and another 20 to compare pix2pixHD samples against Dense-18-growth-32-pix2pixHD.

5.2 Analysis

As shown in Table 2, our proposed modifications had a marginal effect on the inference time of the pix2pixHD generator. Most of our experiments achieved approximately 1% to 2% improvement in inference time compared to the baseline model, except for the DenseNet experiment with 18 dense blocks at a growth rate of 32. Dense-18-growth-32-pix2pixHD ran slighly slower than the baseline, likely due to the larger number of blocks. Both the extra layer experiment and the DenseNet with Spectral Normalization experiment achieved a 2.5% improvement in inference time.

Table 1 compares the results of three test samples on the baseline and on all of our experiments. Although substituting the Residual blocks with Dense blocks improved inference time slightly, the human perceptual study found the DenseNet models produced images of lower quality than the baseline Residual model. In particular, on the Cityscapes dataset, the Residual and Dense networks produced comparable grass, sky, road, and tree textures, but the Residual network produced higher quality buses and cars. We notice that texture quality for small objects declines significantly in DenseNet models. Pedestrians are consistently missed by all DenseNet models, while the baseline residual model produces passable textures. This loss of quality is reflected by the results of our human preference test as shown in Table 3.

Model Type	Mean IoU
pix2pixHD (baseline)	**0.560**
Extra-Layer-pix2pixHD	0.533
Dense-9-growth-32-pix2pixHD	0.501
Dense-9-growth-4-pix2pixHD	0.507
Dense-9-growth-128-pix2pixHD	0.558
Dense-6-growth-32-pix2pixHD	0.523
Dense-18-growth-32-pix2pixHD	0.526
Dense-9-growth-32-spectNorm-pix2pixHD	0.535

Table 4: Mean IoU for each model on the Cityscapes test set.

The Intersection over Union metric in Table 4 supports the human preference results, but does not align precisely. Every DenseNet model reported lower mean Intersection over Union than the baseline pix2pixHD model, although only slightly. However, the DenseNet-9-growth-32 model reported lower IoU than DenseNet-18-growth-32, but better human perception (27% of human preference, versus 19%). The differences between human perception and the IoU metric may be attributed to the IoU calculation method, which relies on the PSPNet to classify and segment objects in the image.

Figure 5: From left to right: the Cityscapes ground truth segmentation map, the PSPNet output for an image generated by pix2pixHD, and the PSPNet output for an image generated by DenseNet-9-growth-128-pix2pixHD.

PSPNet often fails to recognize smaller or thinner objects, like stoplights, as it takes the generated image and creates semantic meaning from them. If smaller objects are not generated well, the PSPNet often interprets the small object as part of the larger semantic object around it, missing out on the finer details. We see this in the baseline, where it misses the top of stoplights, and our DenseNet-9-growth128 model (similar to others), where it misses most of the light post altogether.

Note that we train each of our models only to 60 epochs due to budget constraints, and only on the global generator G_1. Both quantitative and qualitative results could differ after training models for the full 200 epochs as conducted in Wang et al. [2017].

6 Conclusion

Our original goal was to speed up the inference time of the generative compression model in Agustsson et al. [2018]. This required us to increase the inference speed of pix2pixHD. Upon analyzing and understanding the pix2pixHD inference process, we realize that processing the feature maps and developing a one hot encoding of the semantic label takes $1.5\times$ the generation process time. Because this initial pre-processing step is analytical, even large modifications to the pix2pixHD generator cannot improve generator inference time dramatically. However, evaluation shows that our implemented modifications achieved some limited improvements on inference time, albeit suffering some loss in generated image quality.

Our next step is to continue training the experiment models to the full 200 epochs on both the global generator (G_1) and the local generator (G_2), and verify whether the results after 200 epochs remain consistent with our 60-epoch evaluations. In addition, we will perform further analysis on the cause of performance differences between the original pix2pixHD implementation and our various modifications. After achieving satisfactory improvements, we will apply our modified models to the Agustsson et al. [2018] compression architecture and evaluate the performance on the image compression task.

7 Supplemental Materials

7.1 Project Code

Our implementation is available at `https://github.com/nikhilsardana/GANcompression`.

7.2 Demo Video

A short video describing the project is available at `http://bit.do/cs236-video-nnj`.

References

Georges Seguin. *Zoo de la Barben 20100605 048*. Jun 2010. URL `https://commons.wikimedia.org/wiki/File:Zoo_de_la_Barben_20100605_048.jpg`.

Shibani Santurkar, David M. Budden, and Nir Shavit. Generative compression. *CoRR*, abs/1703.01467, 2017. URL `http://arxiv.org/abs/1703.01467`.

Oren Rippel and Lubomir Bourdev. Real-time adaptive image compression. In *International Conference on Machine Learning*, 2017.

Eirikur Agustsson, Michael Tschannen, Fabian Mentzer, Radu Timofte, and Luc Van Gool. Generative adversarial networks for extreme learned image compression. *CoRR*, abs/1804.02958, 2018. URL `http://arxiv.org/abs/1804.02958`.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014. URL `http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf`.

Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. *arXiv preprint arXiv:1711.11585*, 2017.

Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *CoRR*, abs/1411.1784, 2014. URL `http://arxiv.org/abs/1411.1784`.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, volume 1, page 3, 2017.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *CoRR*, abs/1802.05957, 2018. URL `http://arxiv.org/abs/1802.05957`.

Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. *CoRR*, abs/1612.01105, 2016. URL `http://arxiv.org/abs/1612.01105`.

Mohamed Omran Marius Cordts. cityscapes-scripts. `https://github.com/mcordts/cityscapesScripts`, 2018.