HYPE-C: EVALUATING IMAGE COMPLETION MODELS THROUGH STANDARDIZED CROWDSOURCING

Anonymous authors
Paper under double-blind review

ABSTRACT

A significant obstacle to the development of new image completion models is the lack of a standardized evaluation metric that reflects human judgement. Recent work has proposed the use of human evaluation for image synthesis models, allowing for a reliable method to evaluate the visual quality of generated images. However, there does not yet exist a standardized human evaluation protocol for image completion. In this work, we propose such a protocol. We also provide experimental results of our evaluation method applied to many of the current state-of-the-art generative image models and compare these results to various automated metrics. Our evaluation yields a number of interesting findings. Notably, GAN-based image completion models are outperformed by autoregressive approaches.

1 INTRODUCTION

Image completion is a form of image generation in which a set of missing pixels within an image are filled in by a generative model conditioned on the visible pixels. It is is an important problem in machine learning and has a wide range of real-world applications, including photo repair, content-aware filling, and denoising. Image completion also acts as a useful proxy task to measure the progress of generative image models in general, as it is much easier to detect problems such as mode collapse or memorization in image completion models than in image synthesis models.

Over the past few years, an increasing number of image completion models based on GAN and autoregressive architectures have been released. But how do we best evaluate and compare these models? This question is of critical importance for measuring current progress and informing future research. Despite the development of a plethora of new models, there does not yet exist a standardized evaluation method appropriate for the task of image completion.

Reconstruction error metrics–such as l_1 error, l_2 error and peak signal-to-noise ratio (PSNR)–are commonly used, but are a poor method to evaluate such models. There may be many different valid ways to complete an image other than the original content, and these metrics only judge how well a model recreates the missing portion of an image in its original state.

Image synthesis models are often compared using automatic metrics based on deep networks features, such as the Frechét Inception Distance (Heusel et al., 2017) and the Inception Score (Salimans et al., 2016). Such techniques have been adapted to provide alternative metrics applicable to image completion (Zhang et al., 2018), but the use of deep networks to evaluate models is flawed in a number of ways. Not only can deep networks diverge considerably from human perception, as evidenced by their failures on adversarial examples (Nguyen et al., 2015), but the evaluation results can vary substantially depending on the architecture of the evaluation model and the dataset on which it was trained (Barratt & Sharma, 2018).

For these reasons it has been well recognized that there does not yet exist a satisfactory automatic procedure for generative model evaluation–particularly in the case of image completion–and that the gold standard remains human assessment. While it is common to use crowdsourcing platforms such as Amazon Mechanical Turk to evaluate models, these evaluations are usually ad-hoc and do not allow for a direct comparison of different models. Recent work has attempted to standardize human evaluation for image synthesis models and provide an easy to use benchmark. In particular, the HYPE protocol (Zhou et al., 2019) accepts a set of fully synthetic images, randomly mixes them with real images taken from the models training set, and asks humans on crowdsouring platforms to rate each image as real or fake. The HYPE score of a model measures how often the generated images are rated as real by humans.

However, HYPE is not well suited to evaluating image completion models. As it does not provide standardized image masks to define the regions to complete, it is not possible to guarantee a fair and consistent score when applied to image completion. Additionally, due to the fact that HYPE does not define a standardized test set on which to perform completions and selects comparison images randomly from the training set, it is not possible to directly compare models on a per-image basis.

Our two main contributions are as follows. First, we develop a modified version of HYPE, called "HYPE-Completion (HYPE-C)", that resolves these issues and makes it possible to apply HYPE to image completion models. [1] Second, we use HYPE-C to benchmark a number image completion models, well-known GAN architectures which we modify to perform image completion, and autoregressive models which are able to perform image completion out-of-the-box, in order to create a baseline for future work. We evaluate the models at different image resolutions on commonly used datasets, including FFHQ (Karras et al., 2019), CUB (Wah et al., 2011), LSUN-Bedrooms (Yu et al., 2015), and Stanford Cars (Krause et al., 2013). On each dataset, all models are evaluated on the same test split that is disjoint from the training images.

Additionally, our HYPE-C evaluation reveals a number of interesting findings. First, state-of-the-art GAN-based image synthesis models modified to perform image completion often outperform models designed for image completion from the ground up. Second, GAN-based image completion methods are substantially outperformed by autoregressive image completion. Third, high quality image completion remains difficult for all of our evaluated methods across datasets, with the only exception of faces at lower resolutions. Finally, that a model trained to predict human evaluations directly can outperform previous automatic evaluation methods, but is still not a reasonable approximation of human judgement.

2 Related Work

2.1 Per-Pixel Perceptual Metrics

Image completion models are often evaluated using simple perceptual or reconstruction error metrics. These include metrics such as l_1 or l_2 error, as well as peak signal-to-noise ratio (PSNR) and the structural similarity index measure (SSIM). However, while these metrics are well suited to evaluating the quality of lossy compression algorithms or similar tasks, they are unable to provide a reasonable assessment of image completion quality as they simply measure the difference between the original and partially synthetic images at the scale of individual pixels. Under these metrics, an image completion model may receive a poor score despite being able to consistently create realistic and believable images, if the completed regions diverge significantly from the original image in pixel space.

2.2 Automatic Evaluation Metrics for Generative Models

Many methods for evaluating generative image models have been proposed, particularly for unconditional GANs (Borji, 2019). As GANs are prone to overfitting and mode collapse (Arora et al., 2017; Arora & Zhang, 2017), it is important to not only evaluate the visual fidelity of generated images, but also their diversity. The most commonly used automatic evaluation metrics are the Inception Score (Salimans et al., 2016) and the Frechét Inception Distance (FID) (Heusel et al., 2017).

The Inception Score uses an Inception Net model pre-trained on ImageNet to label a large number of generated samples, and measures the KL divergence between the conditional label distribution $p(y \mid x)$ and the marginal distribution $p(y) = p(y \mid x = G(z))dz$. Generated images with a high visual fidelity should be more easily labeled and therefore have a conditional label distribution with low entropy, while the marginal distribution should have a high entropy when the images are diverse. Therefore, the Inception Score $\exp(\mathbb{E}_x[\mathbb{KL}(p(y \mid x) \| p(y))])$ simultaneously evaluates both image

[1] The supporting code used to launch HYPE-C evaluations through Amazon Mechanical Turk will be released publicly upon publication.

quality and diversity. However, the Inception Score may produce deceptive results when applied to models trained on a dataset other than ImageNet (Salimans et al., 2016; Barratt & Sharma, 2018).

The FID uses the same pre-trained Inception Net model, but does not utilize the label distributions. Instead, it embeds a set of real images and a set of generated images into the latent feature space of the Inception Net. Then, modelling the distributions of the embeddings of the real and generated images as multivariate Gaussians $\mathcal{N}(\mu_R, \Sigma_R)$ and $\mathcal{N}(\mu_G, \Sigma_R)$, the FID is the Frechét distance between them. FID is sensitive to both the visual quality of generated images as well as their diversity. However, it makes the strong assumption that the feature vectors follow a Gaussian distribution. Additionally, as in the case of the Inception Score, the FID may not provide accurate results when applied to models trained on datasets other than ImageNet.

Techniques similar to FID have been used for purposes other than GAN evaluation as well. Loss functions computed from the features of a pre-trained network are a key component of neural style transfer (Johnson et al., 2016; Jing et al., 2019) and are frequently used for super-resolution (Johnson et al., 2016; Ledig et al., 2017; Wang et al., 2018). Zhang et al. (2018) recently proposed adapting such techniques to create a perceptual distance metric, which outperforms classical metrics such as PSNR and SSIM in correlation with human evaluations. Although these techniques have produced impressive results in their respective domains, they are still reliant on pre-trained networks, meaning that they may produce vastly different results depending upon the specific network architecture used and the dataset on which the network was trained, and can only act as a rough approximation of how human beings perceive images.

2.3 Human Evaluation Metrics

Direct human evaluation through crowdsourcing platforms such as Amazon Mechanical Turk provides an attractive alternative, as it avoids many of the pitfalls of automatic metrics. The recently produced HYPE (Human eYe Perceptual Evaluation) benchmark (Zhou et al., 2019) standardized the process of performing human evaluations for generative image models, allowing for direct comparison. The first HYPE evaluation method, $\text{HYPE}_{\text{time}}$, tasks human evaluators from Amazon Mechanical Turk with classifying a random sequence of real and fake images. Images are shown for a varying length of time, and the generative model is scored based on the average minimum length of time required for the human evaluators to reliably discern real from fake images. The second evaluation method, HYPE_{∞}, again shows evaluators a random sequence of real and fake images, but does not limit the amount of time that each image is displayed, and scores models based on the percentage of images incorrectly classified by the evaluators (both real and fake).

3 Evaluation Method

Although HYPE provides an effective standardized method to perform human evaluations of image synthesis models, it is not directly applicable to image completion. In order to adapt HYPE to the task of image completion, we must define a standardized set of test images, as well as standardized mask or set of masks to apply to the test images denoting which regions to complete.

For our modified HYPE evaluation protocol, we first select 100 random test images and 100 random comparison images from the dataset. We then mask the bottom half of the test images and use the respective image completion model to replace the masked portion, while leaving the comparison images unchanged. All 200 images are then shown in random order to each of the human evaluators, who classify each image as real or synthetic, and are given an unlimited amount of time to view each image. The model's HYPE-C score is simply the percentage of images (both real and completed) that are misclassified by the human evaluators. In the scenario that the model produces perfect completed images, the human evaluators are no longer able to distinguish the difference between real and partially synthetic images, reaching a HYPE-C score of 50% or more. In our experiments we use 15 human evaluators for each model evaluation. The primary difference between HYPE-C and the original HYPE is the use of a standardized test set and image mask.

Note that while masking the bottom half of each image does not necessarily replicate a real-world use case, it allows us to evaluate the widest range of model architectures, and provides a useful baseline.

Quality Control: To ensure high-quality results, we follow the approach used in HYPE of requiring evaluators to pass a qualification exam. Potential evaluators are shown a random sequence of 50 real and 50 synthetic images, and must correctly classify at least 65 of the given images. Additionally, to encourage workers to remain fully focused on their given tasks, we pay evaluators a bonus of $0.02 for every correctly classified image.

4 EXPERIMENTS

4.1 DATASETS

We trained models on the Flickr-Faces-HQ (FFHQ) dataset (Karras et al., 2019), the Caltech-UCSD Birds-200-2011 (CUB) dataset (Wah et al., 2011), the Stanford Cars dataset (Krause et al., 2013), the LSUN-Bedroom dataset (Yu et al., 2015), and the LSUN-Cat dataset (Yu et al., 2015) at 32x32, 64x4, and 128x128 resolution.

Train and Test Split: From each dataset, we randomly selected 100 test images and 100 comparison images from the dataset's predefined test-set if available. For datasets that did not have a predefined test-set, we generated a new train-validation-test split, with 100 test images and 100 comparison images set aside for HYPE-C evaluation. The same train-validation-test splits were used for the evaluation of all models at all resolutions.

4.2 EVALUATED MODELS

Inpainting Method	Model	Feasible Resolutions
Latent Space Search	StyleGAN (Karras et al., 2019)	32x32, 64x64, 128x128
	ProGAN (Karras et al., 2018)	32x32, 64x64, 128x128
	WGAN-GP (Gulrajani et al., 2017)	32x32, 64x64, 128x128
Conditional GAN	Conditional StyleGAN (Karras et al., 2019)	32x32, 64x64, 128x128
	Conditional ProGAN (Karras et al., 2018)	32x32, 64x64, 128x128
	Conditional WGAN-GP (Gulrajani et al., 2017)	32x32, 64x64, 128x128
	DeepFill (Yu et al., 2020)	32x32, 64x64, 128x128
Autoregressive Generation	PixelCNN++ (Salimans et al., 2019)	32x32, 64x64
	PixelSNAIL (Chen et al., 2018)	32x32, 64x64
	Pixel Constrained CNN (Dupont & Suresha, 2018)	32x32, 64x64

Table 1: Model Architectures

In Table 1 we list all of the tested model architectures. We evaluated a combination of both GAN-based image completion methods and autoregressive methods. The evaluated methods can be divided into three categories:

Latent Space Search: We adapt several unconditional GAN architectures to image completion using the latent space search technique proposed by Yeh et al. (2018). Given any trained GAN model, we search the latent space for an optimal latent vector, one that maximizes the discriminator score of the generated image, while minimizing the difference between the generated and the input image.

Conditional GAN Generation: We also adapt unconditional GAN architecture to perform image completion by modifying the architecture directly. We swap the latent inputs to the generator with features conditioned on the upper-half of the image, while making minimal changes to other parts of the model. More specifically, we adopt an encoder-decoder architecture as the generator, where the encoder consists of multiple convolutional layers, and the decoder is the one used in the original generator. Skip connections are made to propagate the high-resolution details from the encoder to the decoder. In terms of training objectives, in addition to the adversarial term, the generator is also trained with an image reconstruction term that minimizes the difference between the inpainted and the original image (Wang et al., 2020). Additionally, we test the DeepFill architecture (Yu et al., 2020), which is designed for image completion.

Autoregressive Generation: Our evaluation method can in general be directly applied to autoregressive generative models. As PixelCNN++ (Salimans et al., 2019) and PixelSNAIL (Chen et al., 2018) generate images pixel-by-pixel in raster scan order, it is not necessary to modify the model architecture to generate the bottom half of test images. Additionally, Pixel Constrained CNN (Dupont & Suresha, 2018) is an autoregressive model specifically designed for image completion and requires no modification for our evaluation.

4.3 Evaluation Results

Model	FFHQ	Stanford Cars
StyleGAN (Karras et al., 2019)	0.119	0.126
ProGAN (Karras et al., 2018)	0.303	0.159
WGAN-GP (Gulrajani et al., 2017)	0.095	0.164
Conditional StyleGAN (Karras et al., 2019)	0.453	0.160
Conditional ProGAN (Karras et al., 2018)	0.440	0.111
Conditional WGAN-GP (Gulrajani et al., 2017)	0.156	0.135
DeepFill (Yu et al., 2020)	0.259	0.080
PixelCNN++ (Salimans et al., 2019)	0.486	0.306
PixelSNAIL (Chen et al., 2018)	0.458	0.293
Pixel Constrained CNN (Dupont & Suresha, 2018)	0.200	0.166

Table 2: HYPE-C scores of different models on FFHQ and Stanford Cars at 32x32 resolution.

In order to identify and filter out models that are not capable of higher resolution image generation, we first performed an initial set of evaluations of all model architectures on the FFHQ and Stanford Cars datasets at 32x32 resolution, the results of which can be seen in Table 2. Autoregressive methods in particular are often unable to cope with high resolution image generation. On the other hand, autoregression and conditional GAN models achieve the best results. For a qualitative comparison of images completed by the selected models, see Figure 1.

Based on these initial results, we selected four top-performing model architectures (ProGAN, Con-StyleGAN, Con-ProGAN, and PixelCNN++), including at least one model from each category of image inpainting methods. We then evaluated these models on the remaining datasets and resolutions (Table 3). Figure 2, 3 show qualitative examples. More are provided in appendix.

Resolution	Dataset	ProGAN	Con-StyleGAN	Con-ProGAN	PixelCNN++
32x32	FFHQ	0.303	0.453	0.4403	0.486
	Stanford Cars	0.159	0.160	0.111	0.306
	CUB	0.331	0.399	0.377	0.376
	LSUN-Bedroom	0.387	0.310	0.415	0.465
	LSUN-Cat	0.353	0.403	0.407	0.461
64x64	FFHQ	0.123	0.307	0.374	0.369
	Stanford Cars	0.088	0.086	0.138	0.239
	CUB	0.122	0.156	0.303	0.337
	LSUN-Bedroom	0.245	0.303	0.352	0.283
	LSUN-Cat	0.129	0.368	0.317	0.374
128x128	FFHQ	0.077	0.218	0.298	x
	Stanford Cars	0.047	0.084	0.050	x
	CUB	0.106	0.146	0.167	x
	LSUN-Bedroom	0.087	0.214	0.269	x
	LSUN-Cat	0.109	0.252	0.227	x

Table 3: Full results for ProGAN, Con-StyleGAN, Con-ProGAN, and PixelCNN++.

There are a few notable aspects of these results. First, our modified GAN architectures are able to outperform models designed for image completion. Second, PixelCNN++ outperforms all of the other models on every dataset at both 32x32 and 64x64 resolution. Third, even at these low resolutions, the evaluated models perform poorly on all datasets, with the exception of FFHQ at 32x32 resolution. The current state-of-the-art unconditional GAN models are capable of generating realistic images at a resolution of 1024x1024 or higher (Karras et al., 2019; 2018), yet perform poorly when they are tasked with completing half of a low resolution image.

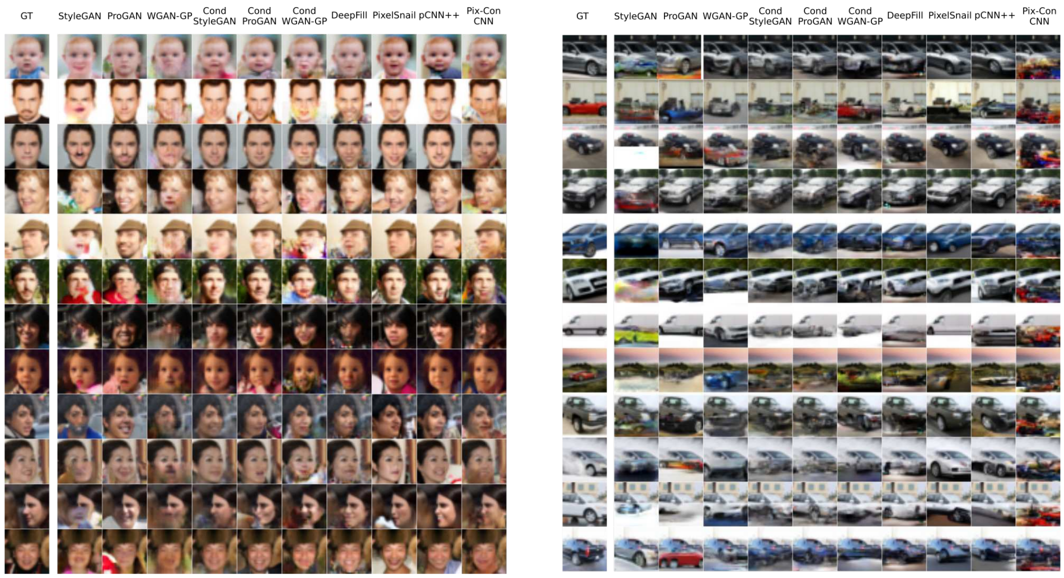

Figure 1: Qualitative comparison between different methods at 32x32 resolution. More results can be found in the appendix.

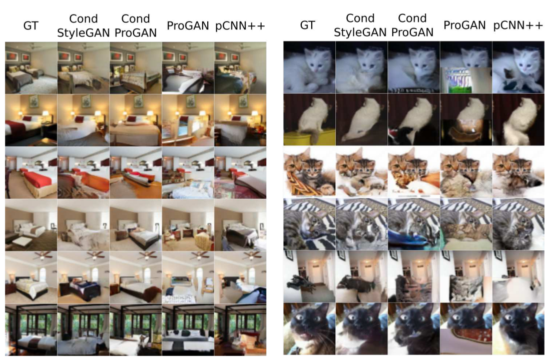

Figure 2: Qualitative results at 64x64 resolution.

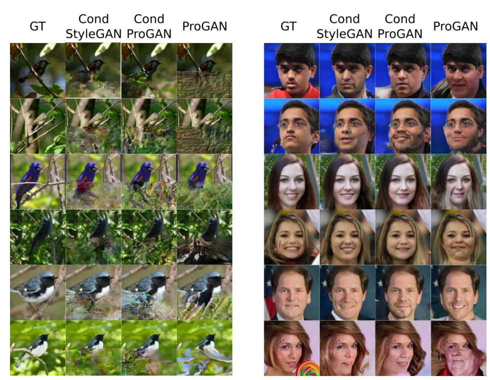

Figure 3: Qualitative results at 128x128 resolution.

4.4 Comparison with Automatic Metrics

We compare FID scores with HYPE-C for each of our evaluated models in Figure 4 in order to determine how well corresponds to human perception. To evaluate an unconditional GAN model, one would usually generate a large set of synthetic images and then compute the FID between that set and the model's training set. Since we are evaluating image completion models, we complete a subset of the training set and compute the FID between it and the full unmodified training set. We complete the same subset of images when evaluating all models at all resolutions.

Since we are performing image completions, we are limited to generating a set of samples of equal size to the training set with which to compute activation statistic. This is particularly problematic with smaller datasets such as Stanford Cars and CUB, where the training set may have only a few thousand images. Due to the increased computational cost of image completion compared to image generation, we also limit the size of our samples to at most 10,000 images per dataset.

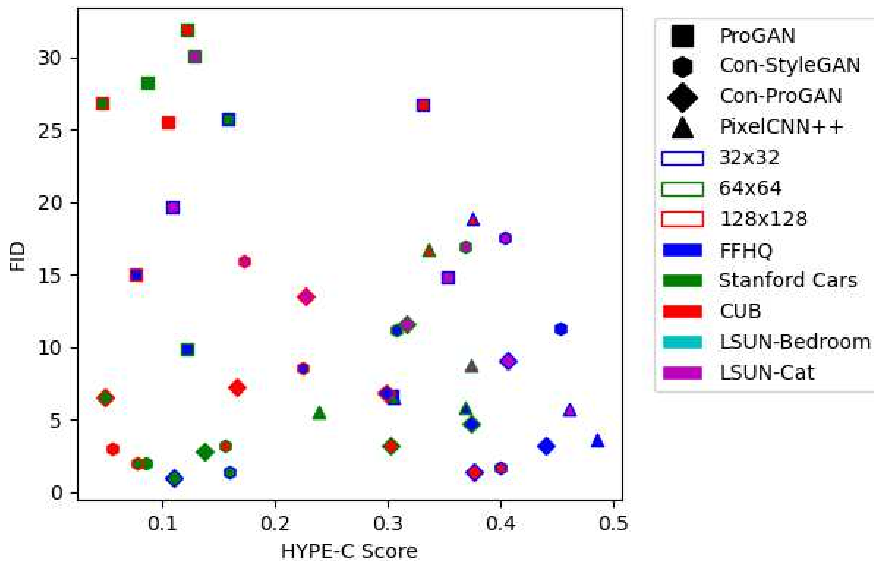

Figure 4: FID vs HYPE-C Scores

It is clear from the figure that the evaluations of model performance using FID and HYPE-C do not coincide. This is not entirely unexpected, as $\text{HYPE}_{\text{time}}$ and HYPE_{∞} scores were shown to be uncorrelated with FID by Zhour et al. (Zhou et al., 2019). However, FID measures both image quality and diversity, while $\text{HYPE}_{\text{time}}$ and HYPE_{∞} only measure image quality, making it somewhat of an unfair comparison. Since HYPE-C evaluates models based on their ability to complete a set of images representative of the target distribution, it is sensitive to image diversity, and we can much more directly compare our results with FID.

We use Spearman rank correlation coefficients to determine how strongly FID correlates with HYPE-C scores across all models at each resolution. We also measure the correlation between the human evaluations on individual images and four different perceptual metrics – perceptual distance (Zhang et al., 2018), peak signal-to-noise ratio (PSNR), the structural similarity index (SSIM), and mean squared error (MSE). The results can be found in Table 4. A value of $\rho = \pm 1$ indicates a perfect positive or negative correlation, while a value of $\rho = 0$ indicates no correlation. We see that FID is nearly completely uncorrelated with HYPE-C, while the perceptual metrics achieve no more than a weak correlation with human evaluations.

Metric	32x32		64x64		128x128	
	ρ	p	ρ	p	ρ	p
FID	-0.003	0.991	-0.052	0.850	-0.028	0.931
Perceptual Distance	-0.342	0.00	-0.322	0.00	-0.306	0.00
PSNR	0.133	0.00	0.196	0.00	0.240	0.00
SSIM	0.278	0.00	0.373	0.00	0.326	0.00
MSE	-0.133	0.00	-0.196	0.00	-0.240	0.00

Table 4: Spearman Rank Correlation Coefficients of human evaluations and automatic metrics

4.5 HUMAN SCORE PREDICTOR

We attempt to create an automatic evaluation metric in a way similar to the perceptual distance metric (Zhang et al., 2018). We created training, validation, and test sets using the full set of completed image from our earlier experiments labeled by their individual average human HYPE-C score. We then trained multiple model architectures to predict the human ratings of the completed images.

We tested a variety of core CNN architectures, including Inception Net, AlexNet, ResNet, and VGG. For each architecture, we replaced the final classification layer with a small fully connected network. To prevent over-training, we used early stopping triggered by a stalled validation score. We tested one version of each model where the core CNN weights are held constant, and one where the core CNN is fine-tuned. The results can be found in Table 5 and Table 6.

Model	l_1 Error	ρ	p
Inception Net V3	0.145	0.764	0.00
AlexNet	0.186	0.637	0.00
ResNet-18	0.158	0.728	0.00
ResNet-34	0.143	0.766	0.00
ResNet-50	0.141	0.778	0.00
ResNet-101	0.136	0.784	0.00
ResNet-152	0.132	0.795	0.00
VGG-11	0.183	0.642	0.00
VGG-13	0.183	0.631	0.00
VGG-16	0.179	0.648	0.00
VGG-19	0.182	0.640	0.00

Table 5: Score prediction using pre-trained networks as feature extractors.

Model	l_1 Error	ρ	p
Inception Net V3	0.142	0.773	0.00
AlexNet	0.162	0.704	0.00
ResNet-18	0.139	0.783	0.00
ResNet-34	0.139	0.772	0.00
ResNet-50	0.140	0.771	0.00
ResNet-101	0.141	0.774	0.00
ResNet-152	0.138	0.777	0.00
VGG-11	0.143	0.764	0.00
VGG-13	0.160	0.710	0.00
VGG-16	0.207	0.523	0.00
VGG-19	0.215	0.517	0.00

Table 6: Score prediction with fine-tuning of feature extraction networks.

While these results are a substantial improvement over the automatic metrics we have previously discussed, our score predictor models are not necessarily sufficient to act as a replacement for evaluation by real humans. They will also be vulnerable to the same issues as other metrics based on neural network features, namely that they may produce deceiving results when applied to other datasets, and may have certain ”blindspots” with respect to specific images i.e. adversarial examples.

5 CONCLUSION

In this work we introduced HYPE-C, a modified form of HYPE capable of evaluating image completion models. We provided both qualitative and quantitative experimental results of HYPE-C applied to a variety of image completion models, forming a baseline for comparison. We showed that well-known GAN-based image synthesis models modified to perform image completion can outperform more complex methods in our setting and that autoregressive models can often outperform GANs in terms of human evaluation at low resolutions. Using our evaluation method, we were able to perform an analysis of the efficacy of automatic evaluation metrics, and show that they only weakly correlate with human evaluations. Finally, we evaluated the use of features extracted from a variety of pre-trained networks as a means to create a proxy for human evaluation.

REFERENCES

Sanjeev Arora and Yi Zhang. Do GANs actually learn the distribution? An empirical study. pp. 1–11, 2017. URL http://arxiv.org/abs/1706.08224.

Sanjeev Arora, Rong Ge, Yingyu Liang, Tengyu Ma, and Yi Zhang. Generalization and equilibrium in generative adversarial nets (GANs). *34th International Conference on Machine Learning, ICML 2017*, 1:322–349, 2017.

Shane Barratt and Rishi Sharma. A Note on the Inception Score. 2018. URL http://arxiv.org/abs/1801.01973.

Ali Borji. Pros and cons of GAN evaluation measures. *Computer Vision and Image Understanding*, 179:41–65, 2019. ISSN 1090235X. doi: 10.1016/j.cviu.2018.10.009.

Xi Chen, Nikhil Mishra, Mostafa Rohaninejad, and Pieter Abbeel. PixelSNAIL: An improved autoregressive generative model. *35th International Conference on Machine Learning, ICML 2018*, 2:1364–1372, 2018. URL https://github.com/neocxi/pixelsnail-public. http://arxiv.org/abs/1712.09763.

Emilien Dupont and Suhas Suresha. Probabilistic Semantic Inpainting with Pixel Constrained CNNs. 89, 2018. URL http://arxiv.org/abs/1810.03728.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein GANs. In I Guyon, U V Luxburg, S Bengio, H Wallach, R Fergus, S Vishwanathan, and R Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 2017-Decem, pp. 5768–5778. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/7159-improved-training-of-wasserstein-gans.pdf.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. *Advances in Neural Information Processing Systems*, 2017-Decem(Nips):6627–6638, 2017. ISSN 10495258.

Yongcheng Jing, Yezhou Yang, Zunlei Feng, Jingwen Ye, Yizhou Yu, and Mingli Song. Neural style transfer: A review. *IEEE transactions on visualization and computer graphics*, 2019.

Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pp. 694–711. Springer, 2016.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, pp. 1–26, 2018.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019-June:4396–4405, 2019. ISSN 10636919. doi: 10.1109/CVPR.2019.00453.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3D Object Representations for Fine-Grained Categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013.

Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4681–4690, 2017.

Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 07-12-June-2015:427–436, 2015. ISSN 10636919. doi: 10.1109/CVPR.2015.7298640.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, pp. 2234–2242, 2016.

Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P. Kingma. PixelCNN++: Improving the PixelCnn with discretized logistic mixture likelihood and other modifications. *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, 2019. URL `http://arxiv.org/abs/1701.05517`.

C Wah, S Branson, P Welinder, P Perona, and S Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical report, 2011.

Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 0–0, 2018.

Yi Wang, Ying-Cong Chen, Xiangyu Zhang, Jian Sun, and Jiaya Jia. Attentive normalization for conditional image generation. *arXiv preprint arXiv:2004.03828*, 2020.

Raymond A. Yeh, Teck Yian Lim, Chen Chen, Alexander G. Schwing, Mark Hasegawa-Johnson, and Minhn Do. Image Restoration with Deep Generative Models. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2018-April:6772–6776, 2018. ISSN 15206149. doi: 10.1109/ICASSP.2018.8462317.

Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. LSUN: Construction of a Large-scale Image Dataset using Deep Learning with Humans in the Loop. *arXiv preprint arXiv:1506.03365*, 2015.

Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas Huang. Free-Form Image Inpainting With Gated Convolution. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4470–4479, 2020. doi: 10.1109/iccv.2019.00457.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018.

Sharon Zhou, Mitchell L. Gordon, Ranjay Krishna, Austin Narcomey, Durim Morina, and Michael S. Bernstein. Hype: Human eye perceptual evaluation of generative models. *Deep Generative Models for Highly Structured Data, DGS@ICLR 2019 Workshop*, (NeurIPS), 2019.

A Qualitative and Automatic Metric Results

A.1 FFHQ

A.1.1 32x32

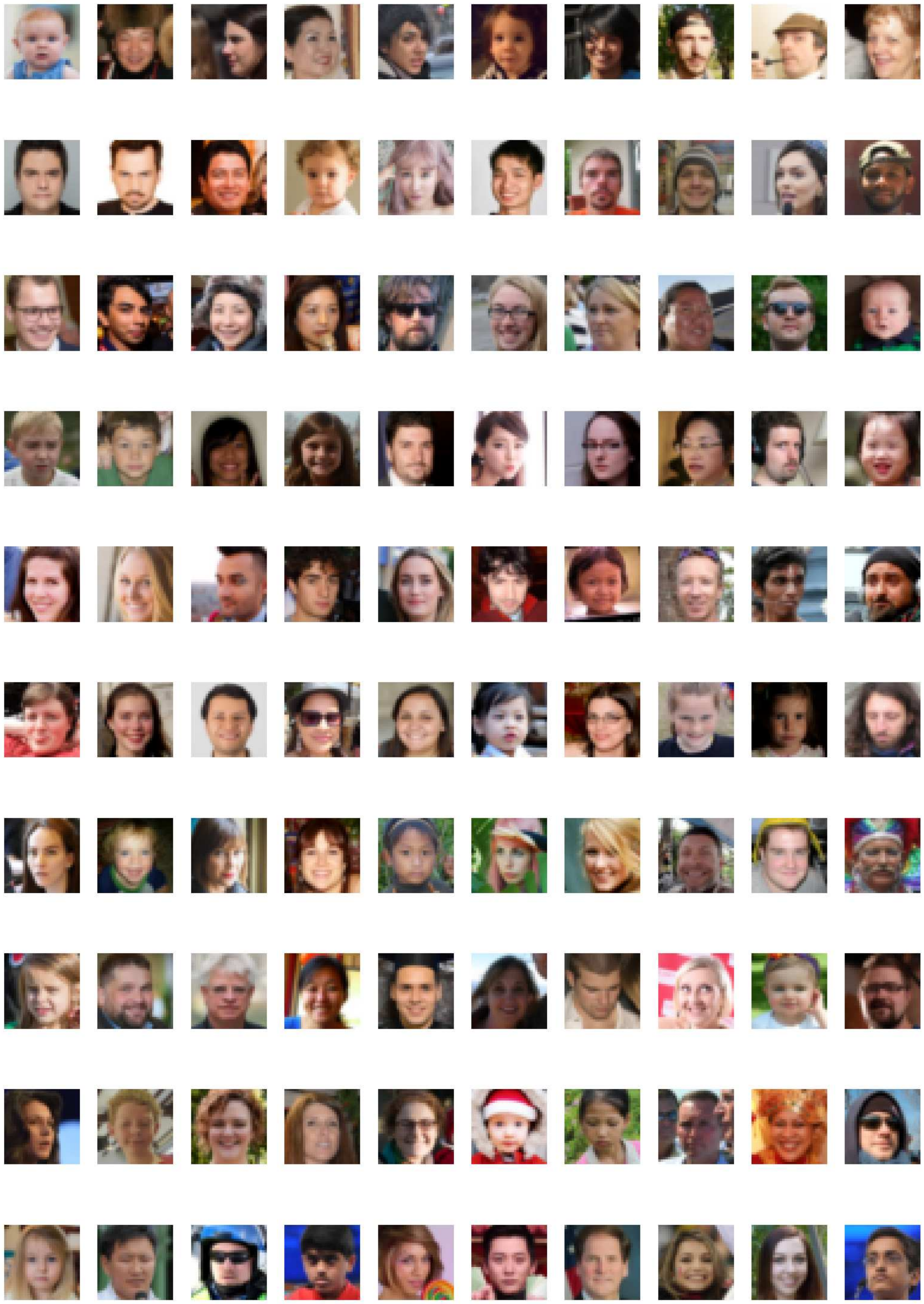

Figure 5: Ground truth for FFHQ at 32x32 resolution.

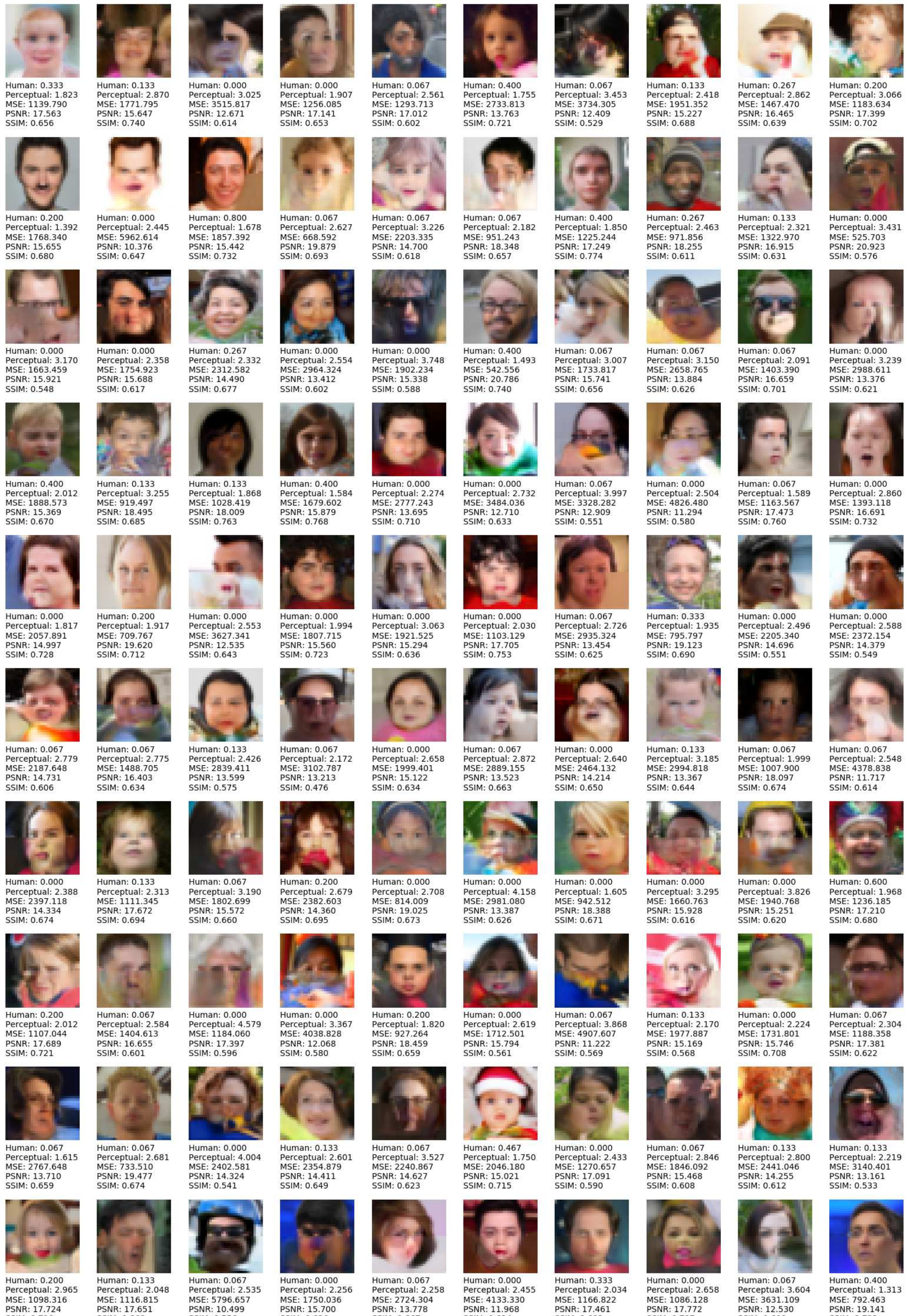

Figure 6: FFHQ 32x32 results for StyleGAN.

Figure 7: FFHQ 32x32 results for ProGAN.

Figure 8: FFHQ 32x32 results for WGAN-GP.

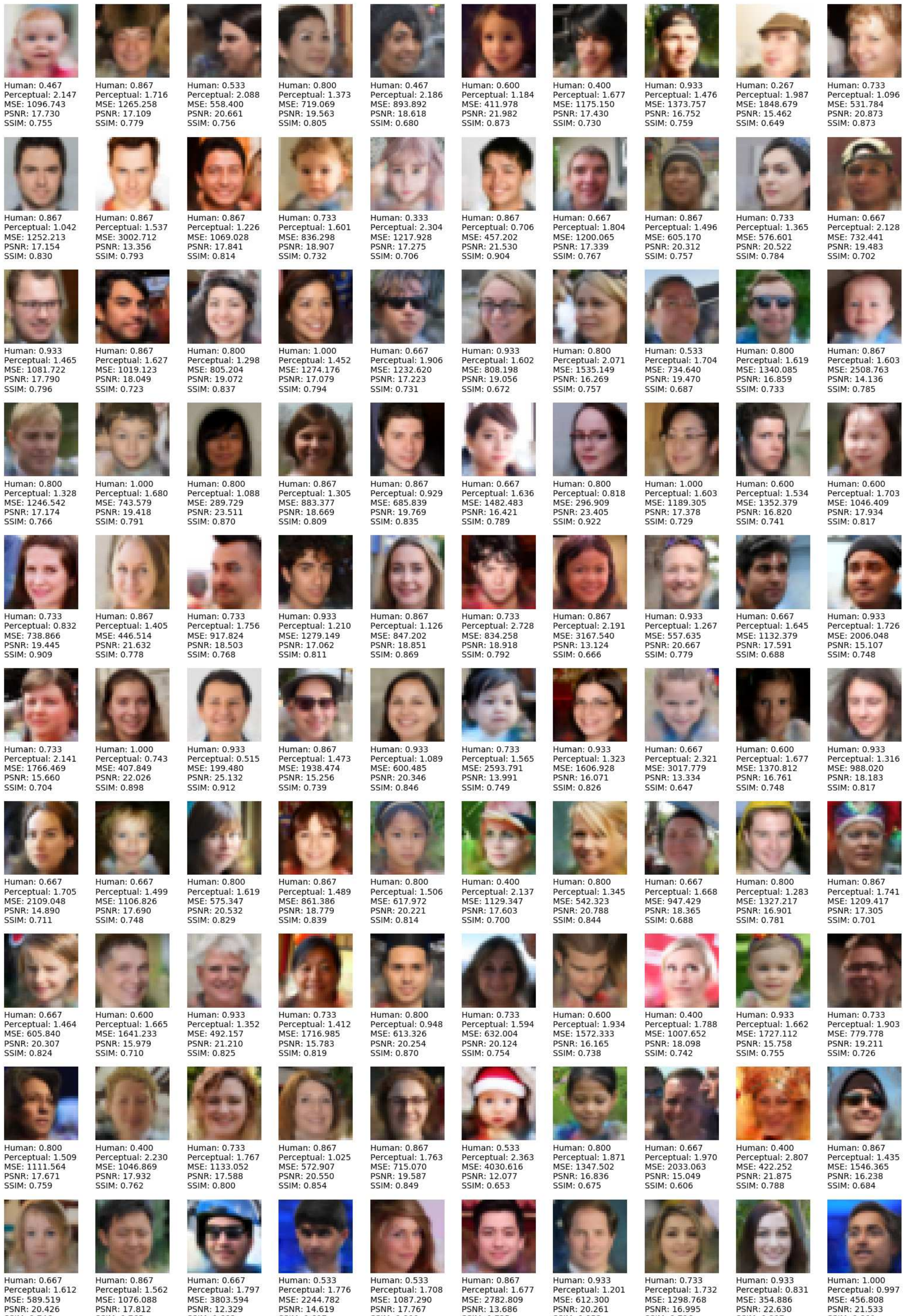

Figure 9: FFHQ 32x32 results for Conditional StyleGAN.

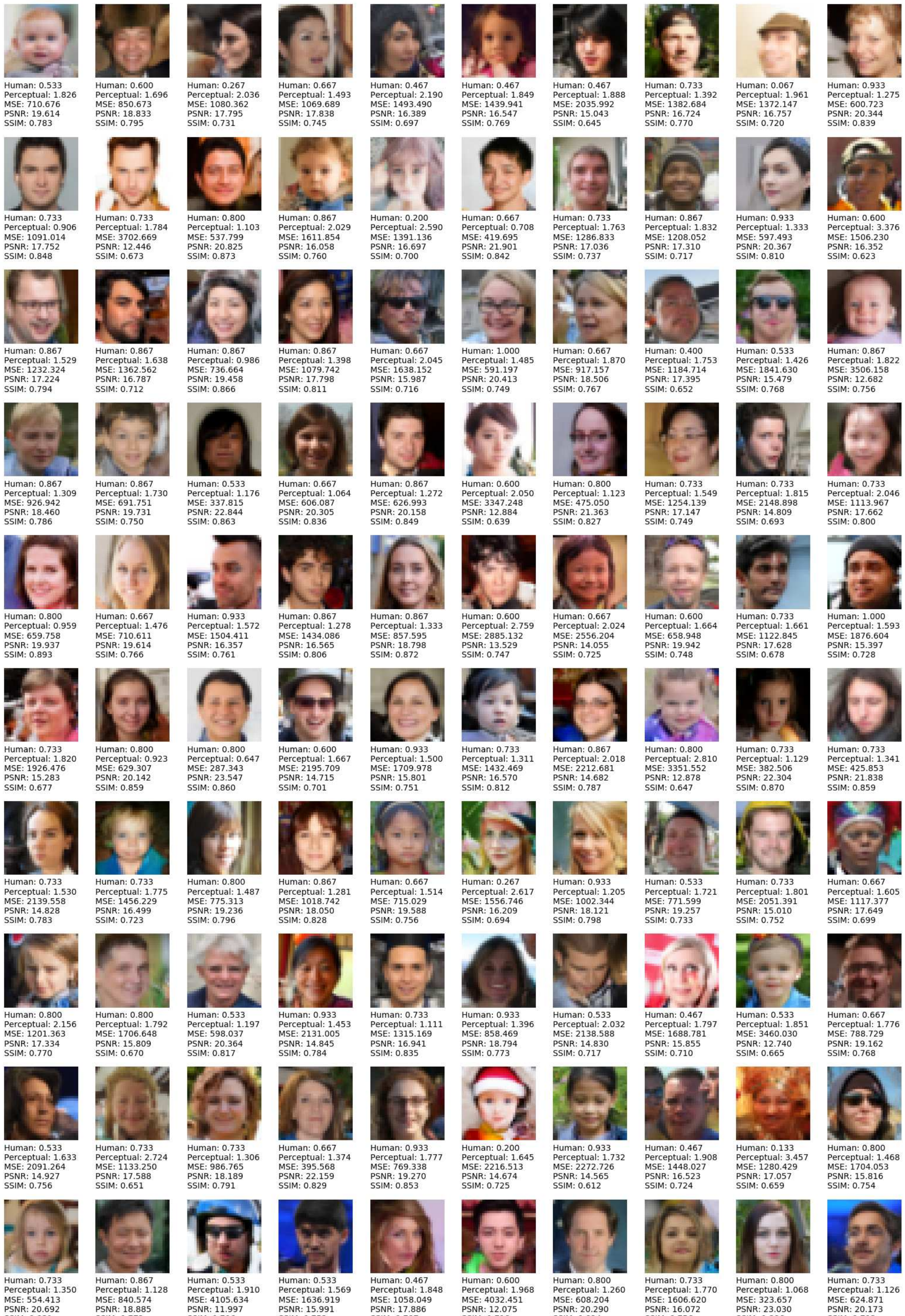

Figure 10: FFHQ 32x32 results for Conditional ProGAN.

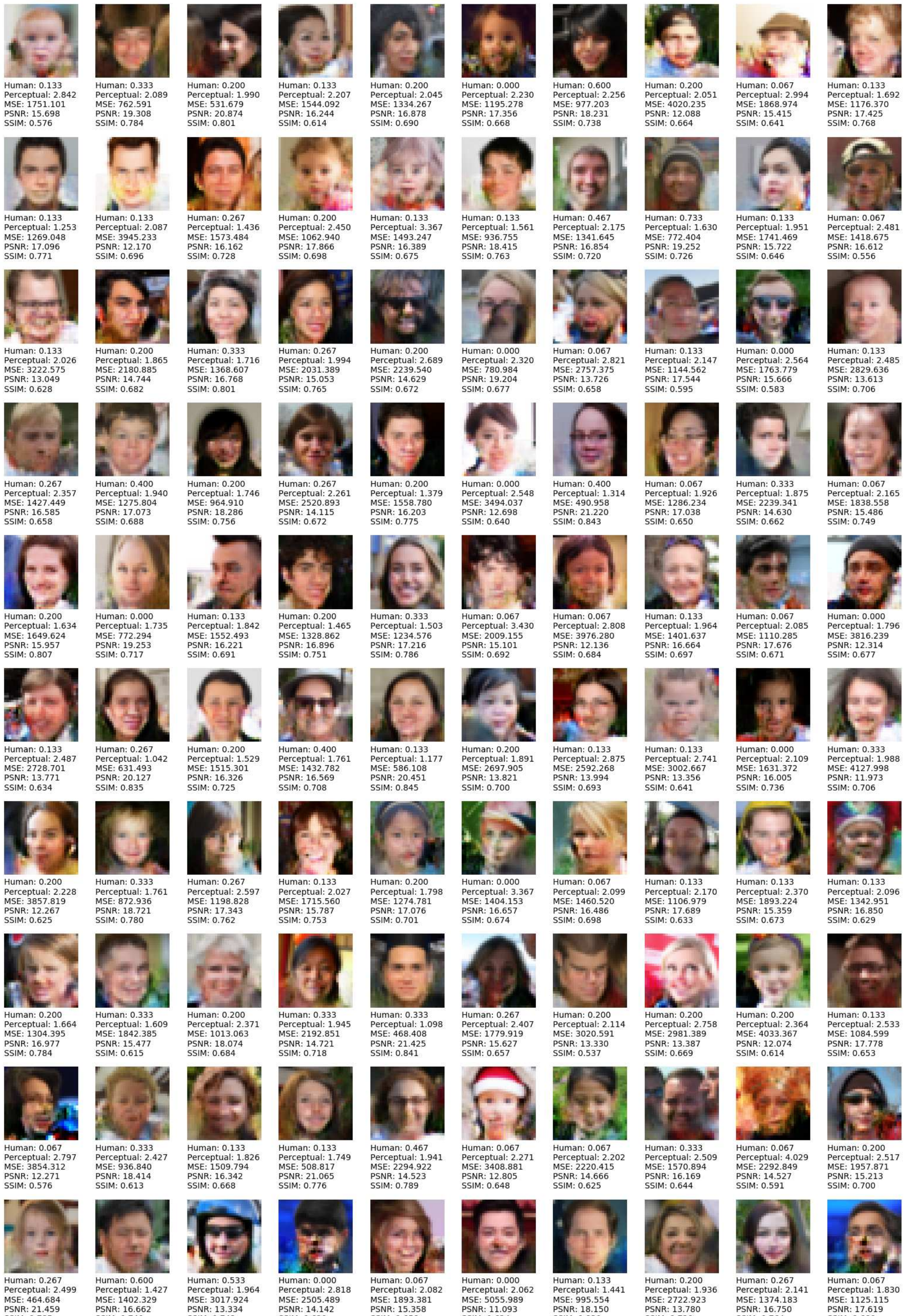

Figure 11: FFHQ 32x32 results for Conditional WGAN-GP.

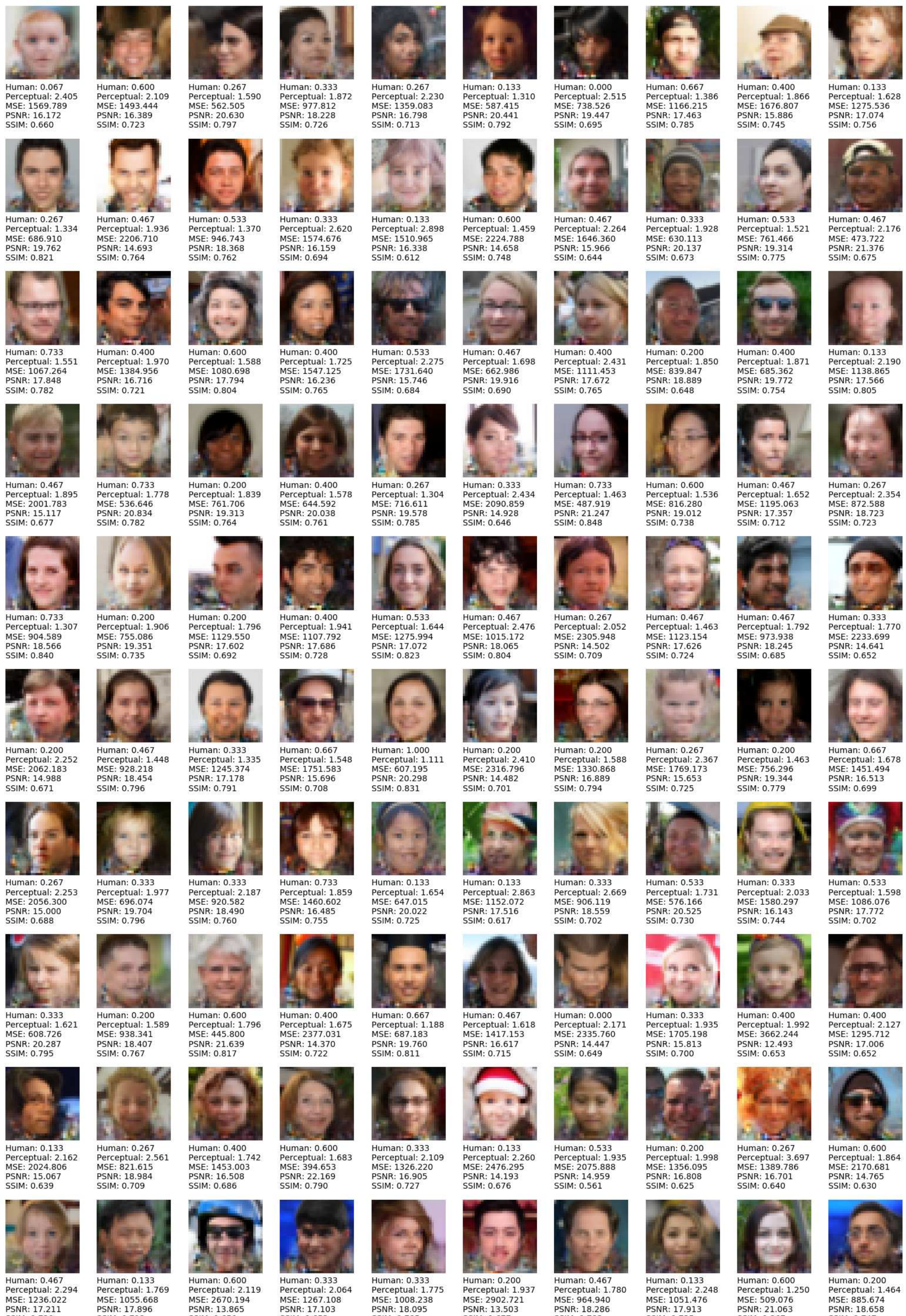

Figure 12: FFHQ 32x32 results for DeepFill.

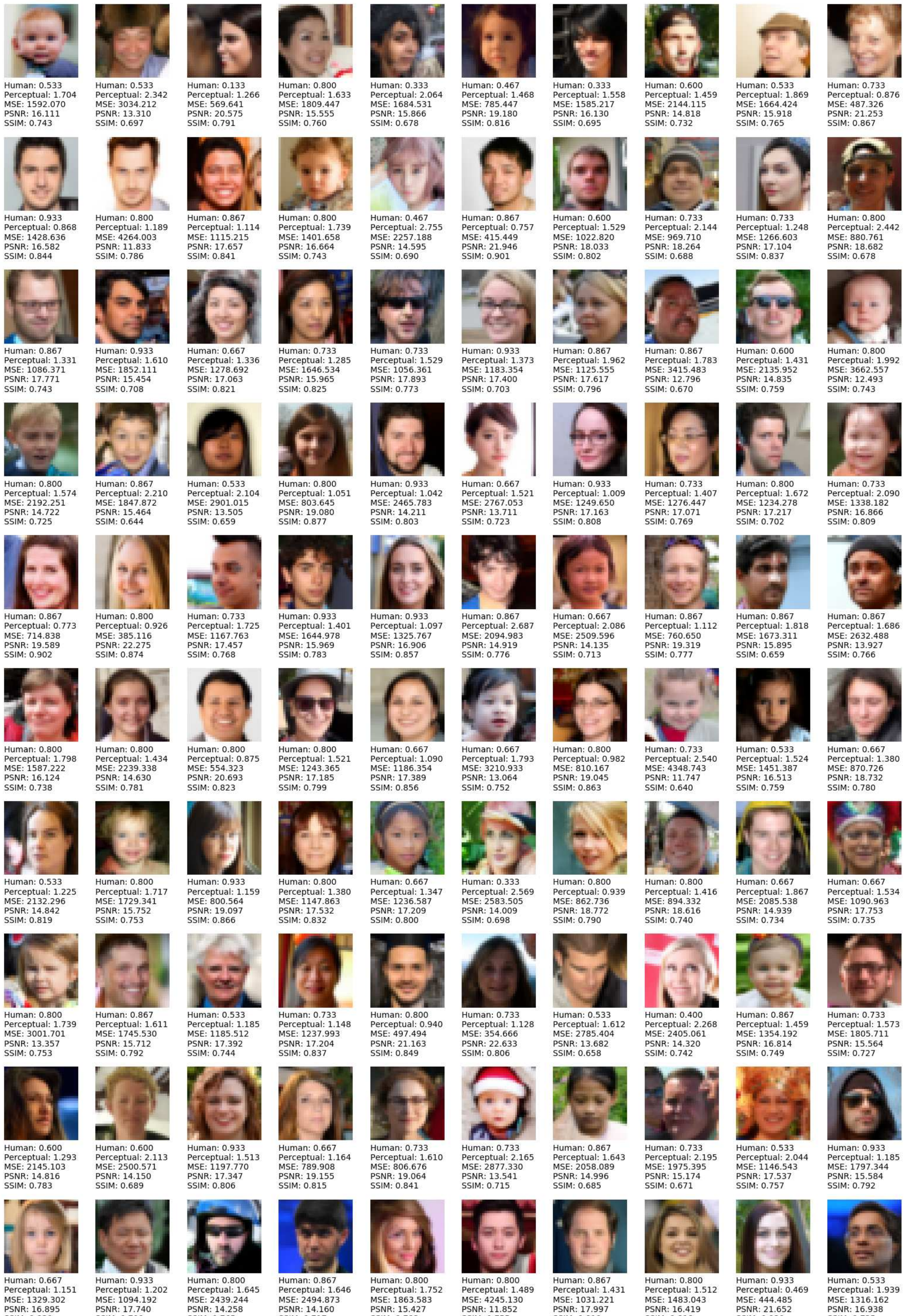

Figure 13: FFHQ 32x32 results for PixelCNN++.

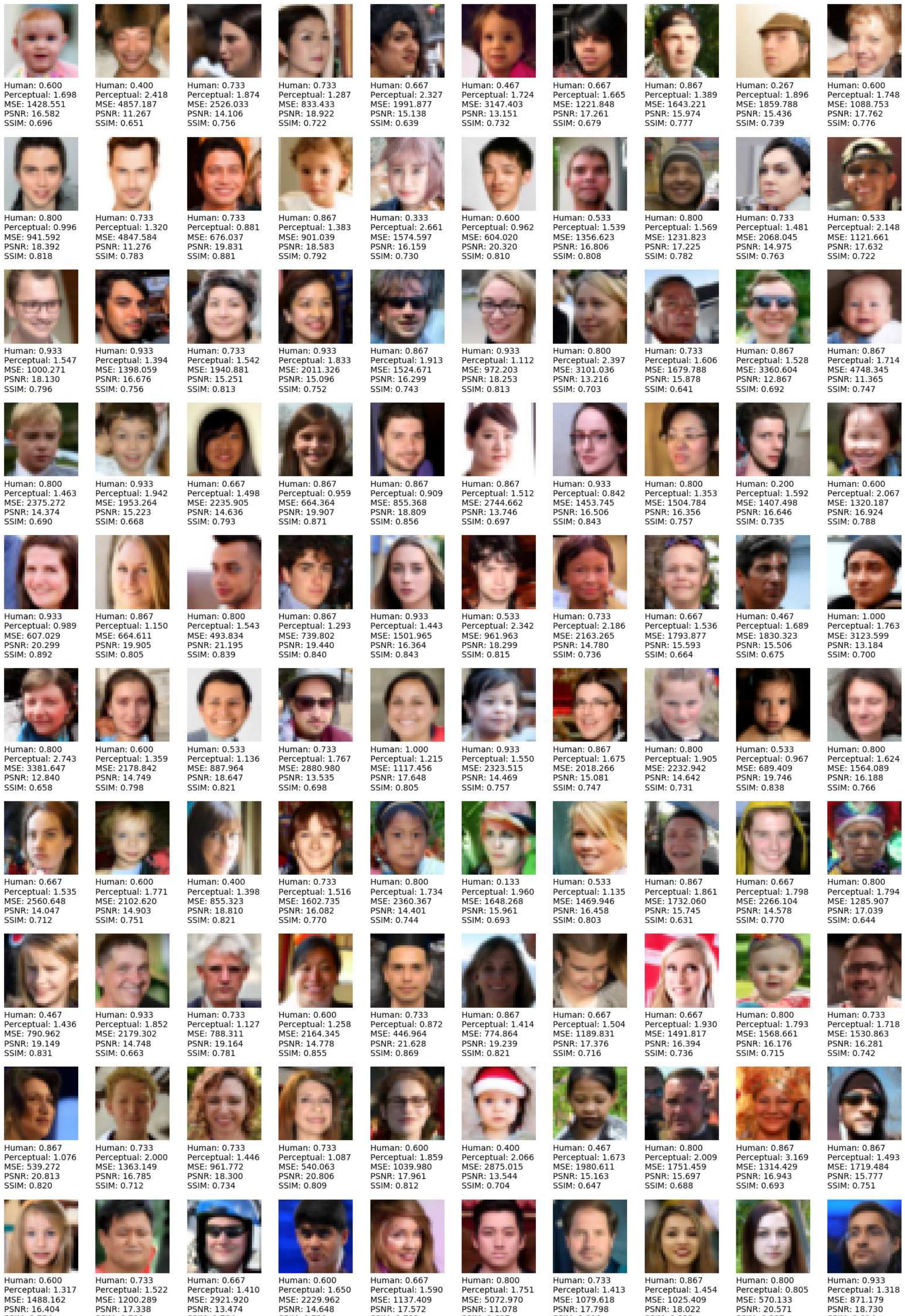

Figure 14: FFHQ 32x32 results for PixelSNAIL.

Figure 15: FFHQ 32x32 results for Pixel Constrained CNN.

A.1.2 64x64

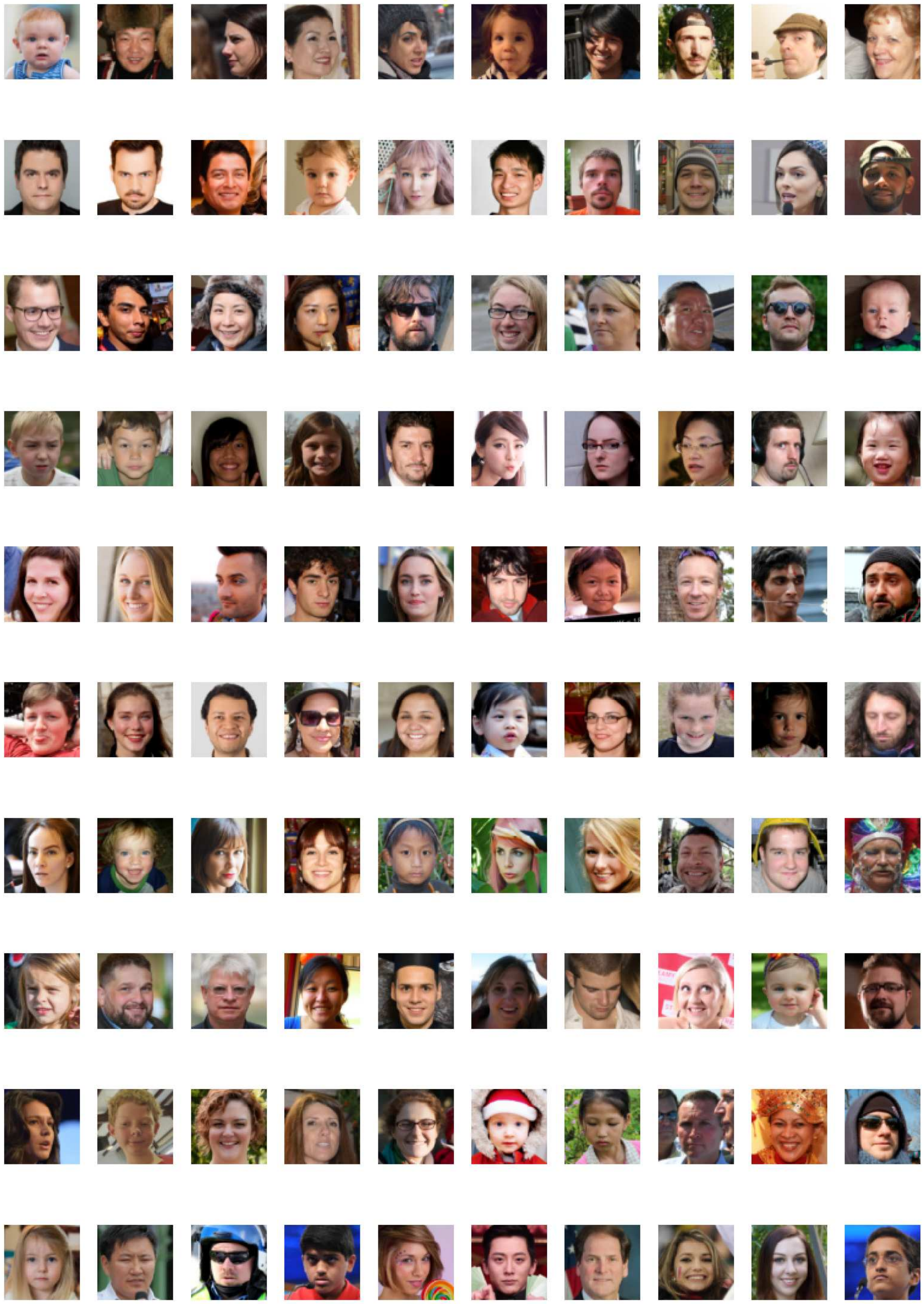

Figure 16: Ground truth for FFHQ at 64x64 resolution.

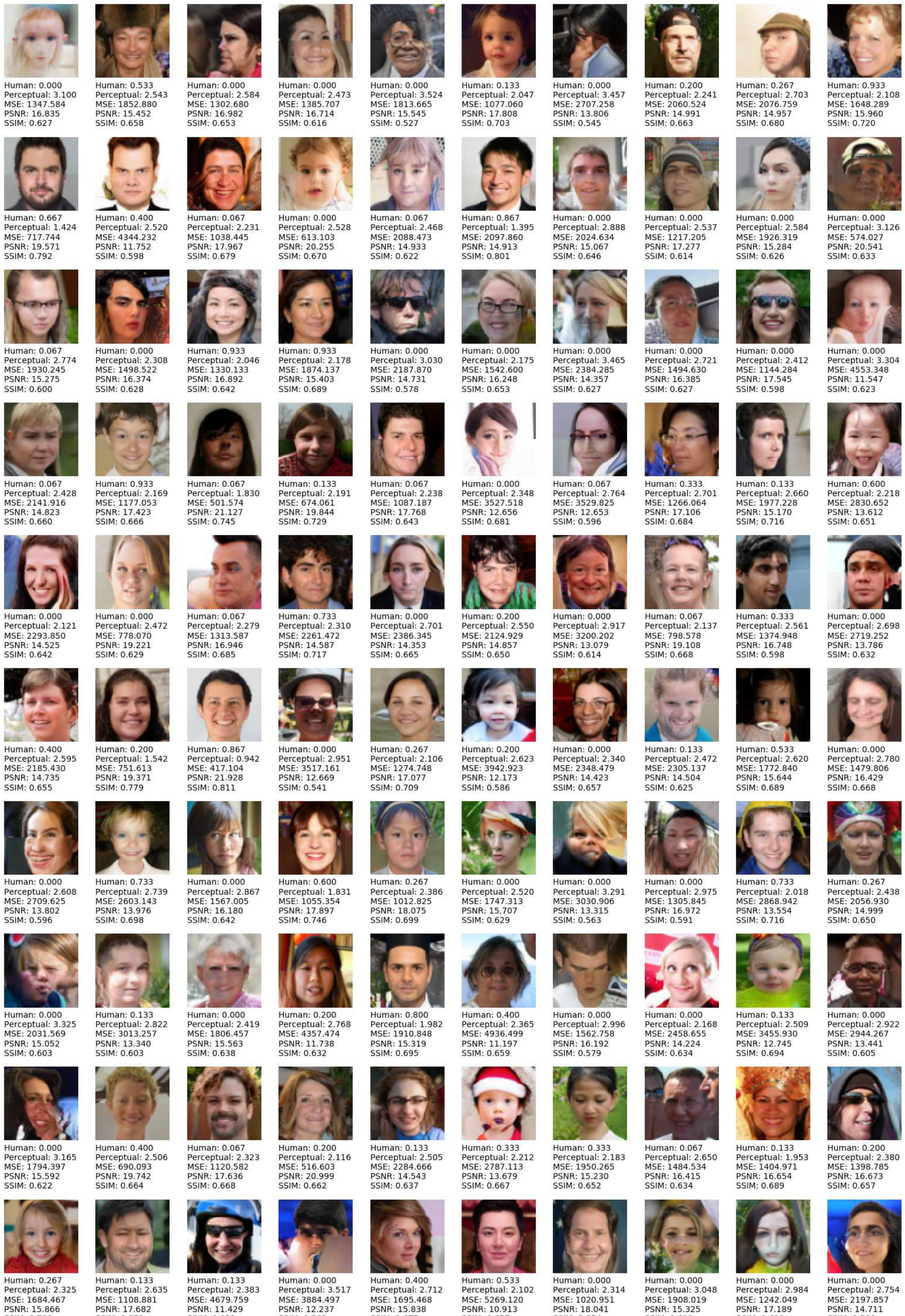

Figure 17: FFHQ 64x64 results for ProGAN.

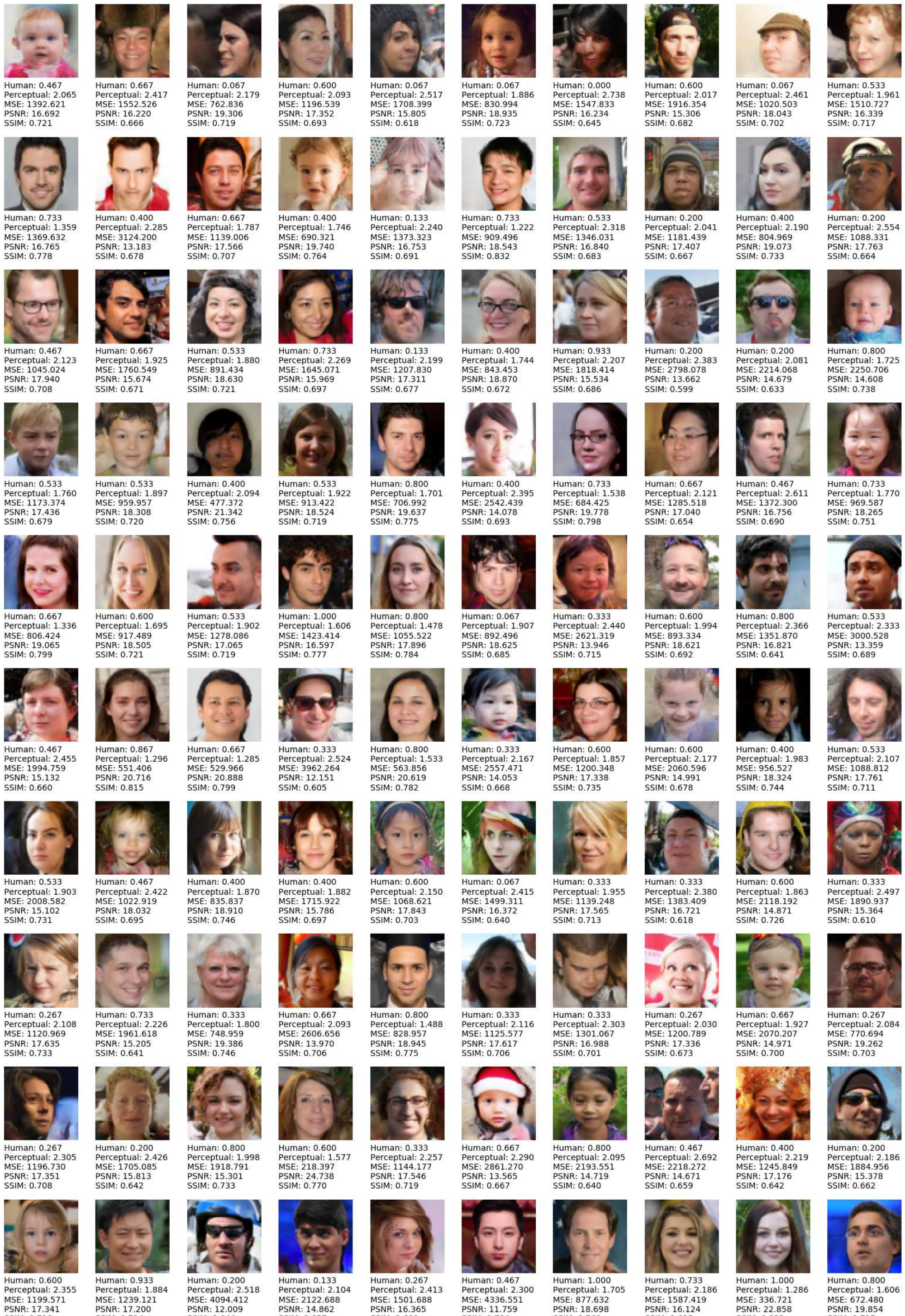

Figure 18: FFHQ 64x64 results for Conditional StyleGAN.

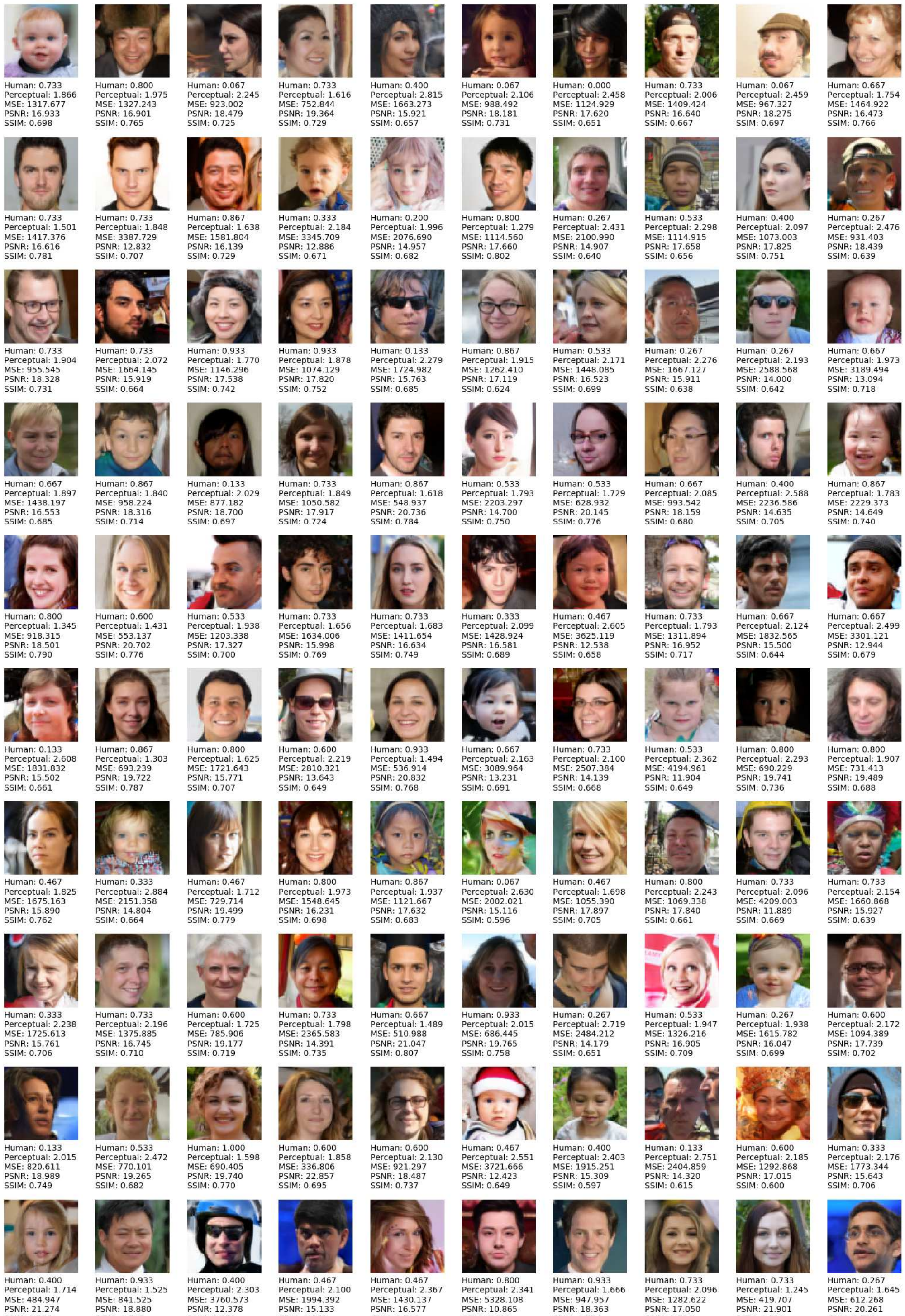

Figure 19: FFHQ 64x64 results for Conditional ProGAN.

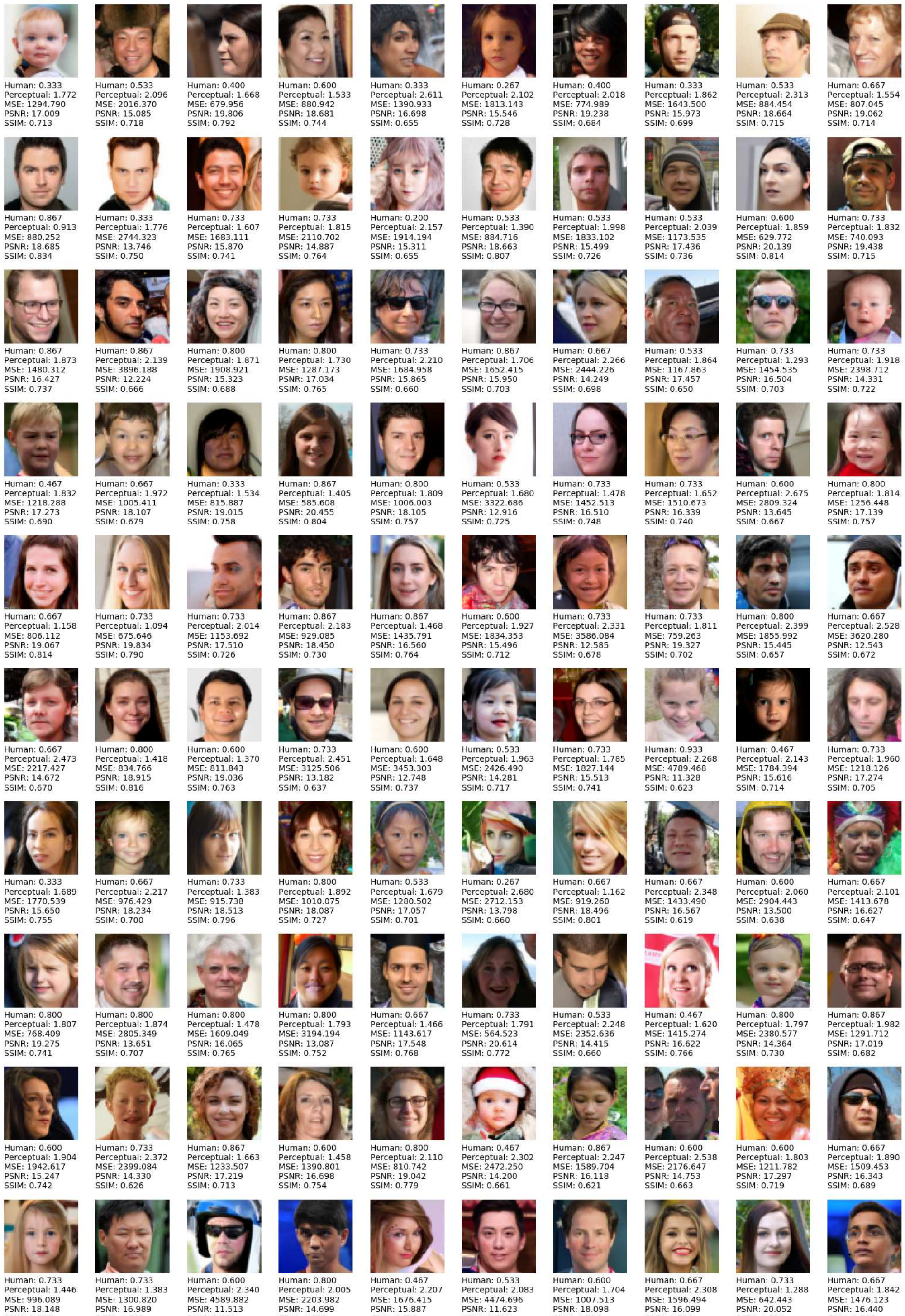

Figure 20: FFHQ 64x64 results for PixelCNN++.

A.1.3 128x128

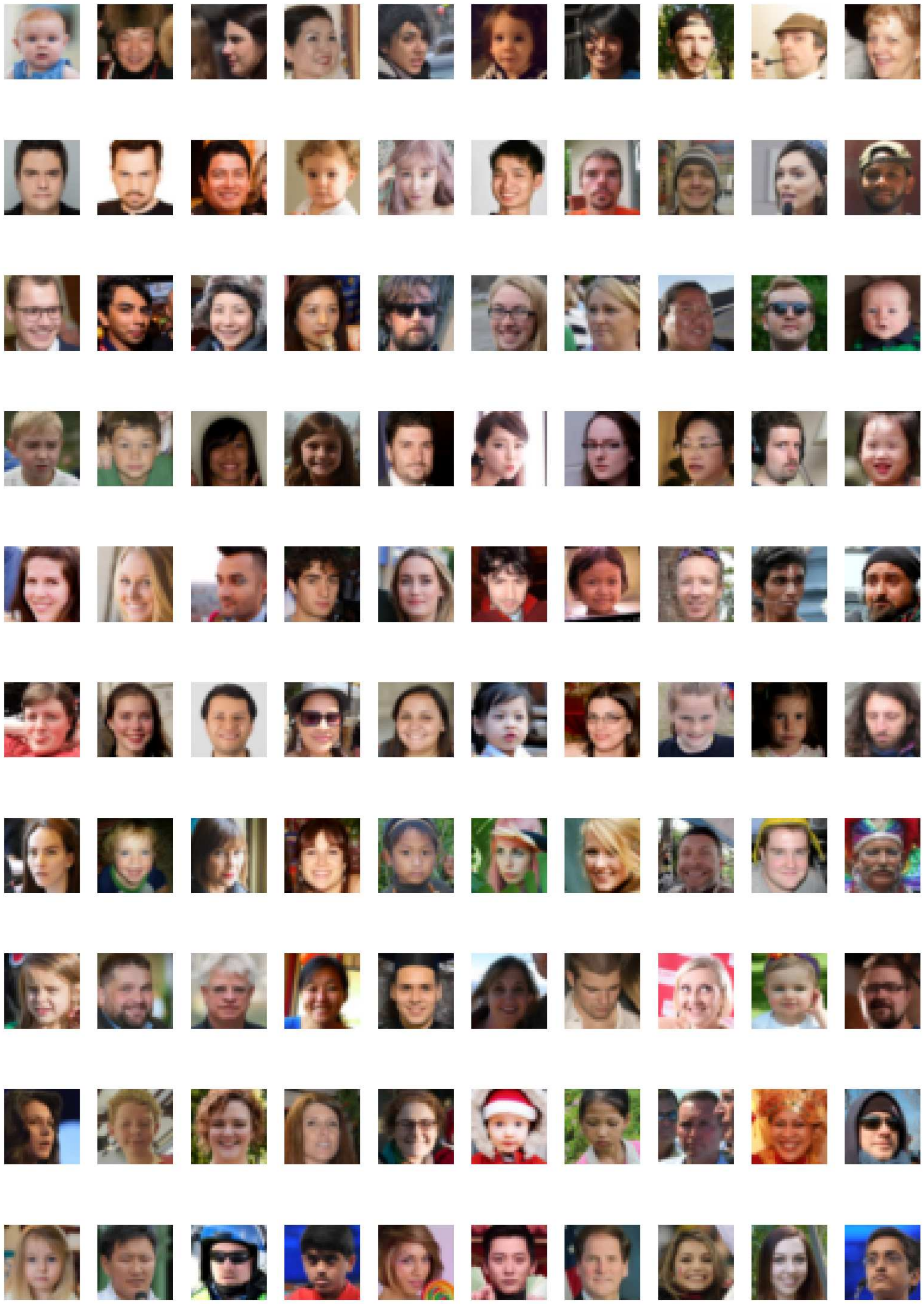

Figure 21: Ground truth for FFHQ at 128x128 resolution.

Figure 22: FFHQ 128x128 results for ProGAN.

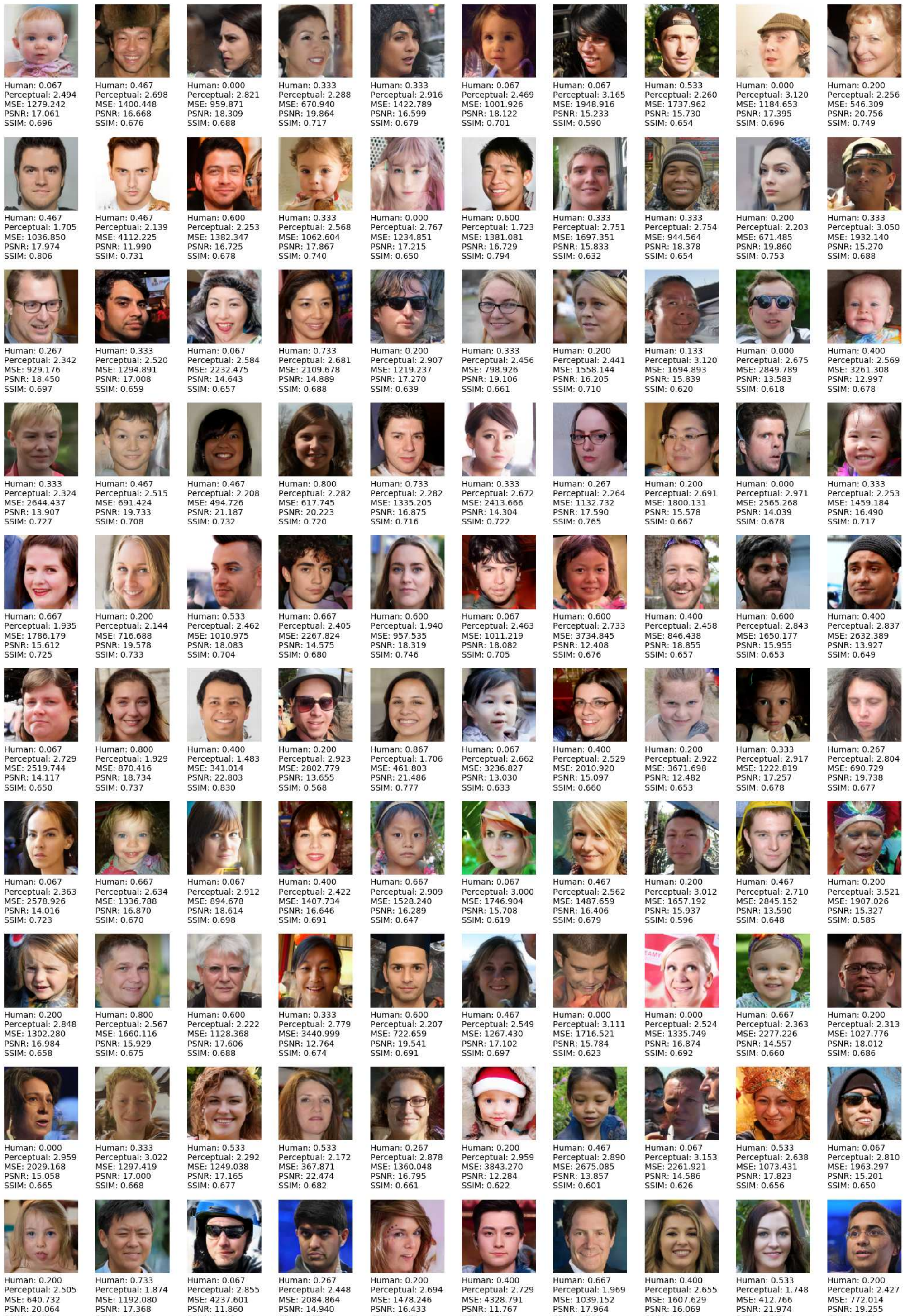

Figure 23: FFHQ 128x128 results for Conditional StyleGAN.

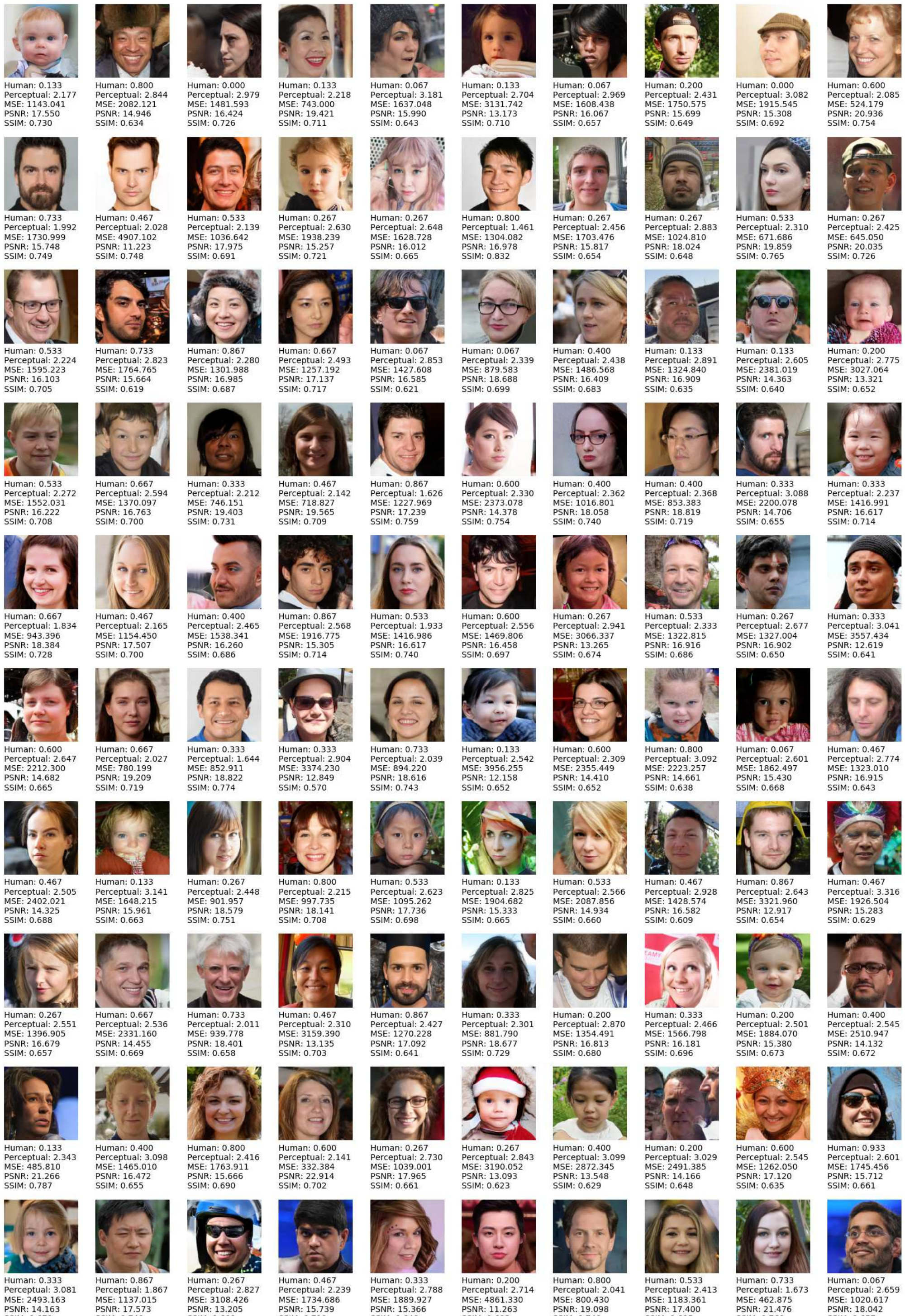

Figure 24: FFHQ 128x128 results for Conditional ProGAN.

A.2 STANFORD CARS

A.2.1 32x32

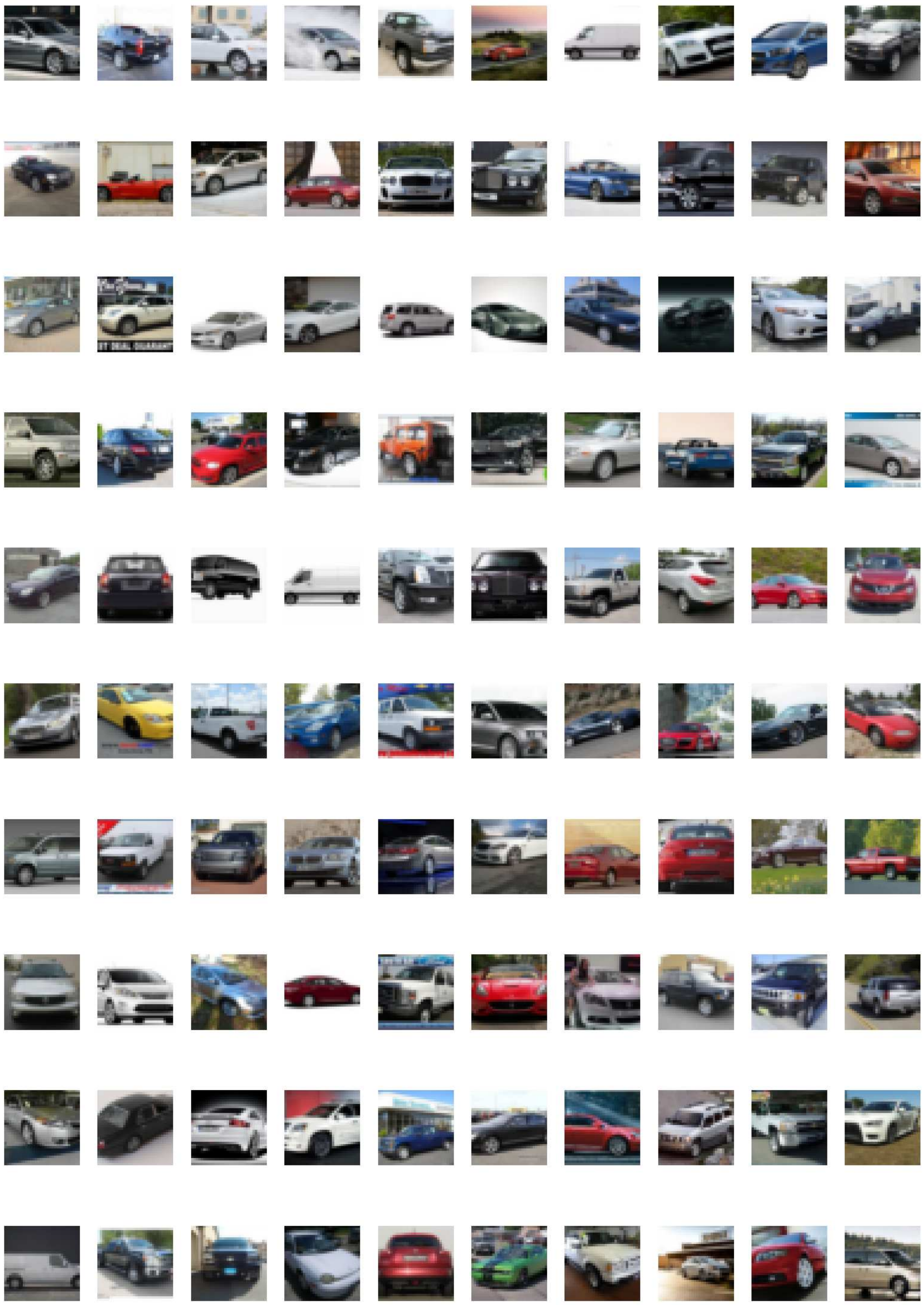

Figure 25: Ground truth for Stanford Cars at 32x32 resolution.

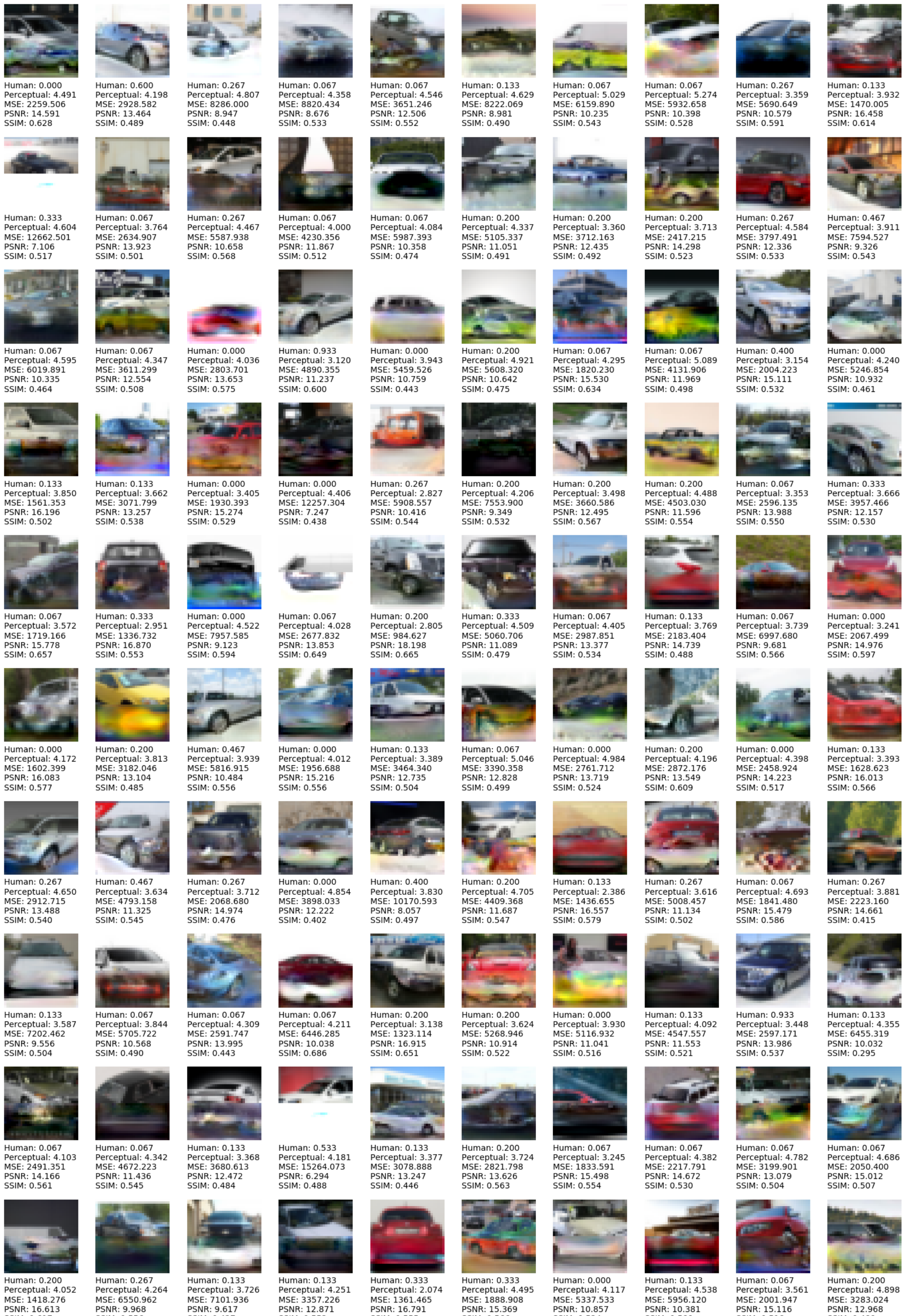

Figure 26: Stanford Cars 32x32 results for StyleGAN.

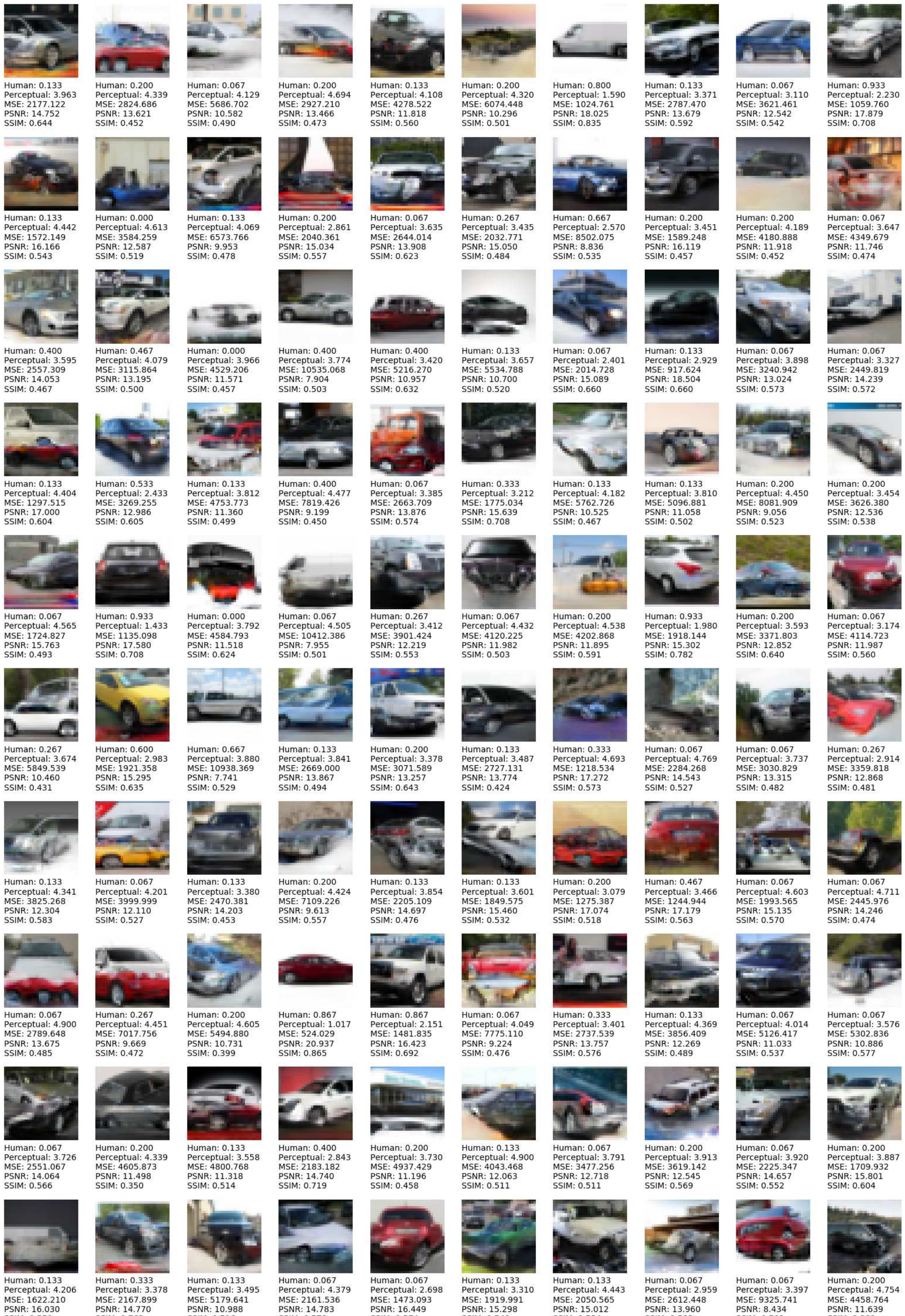

Figure 27: Stanford Cars 32x32 results for ProGAN.

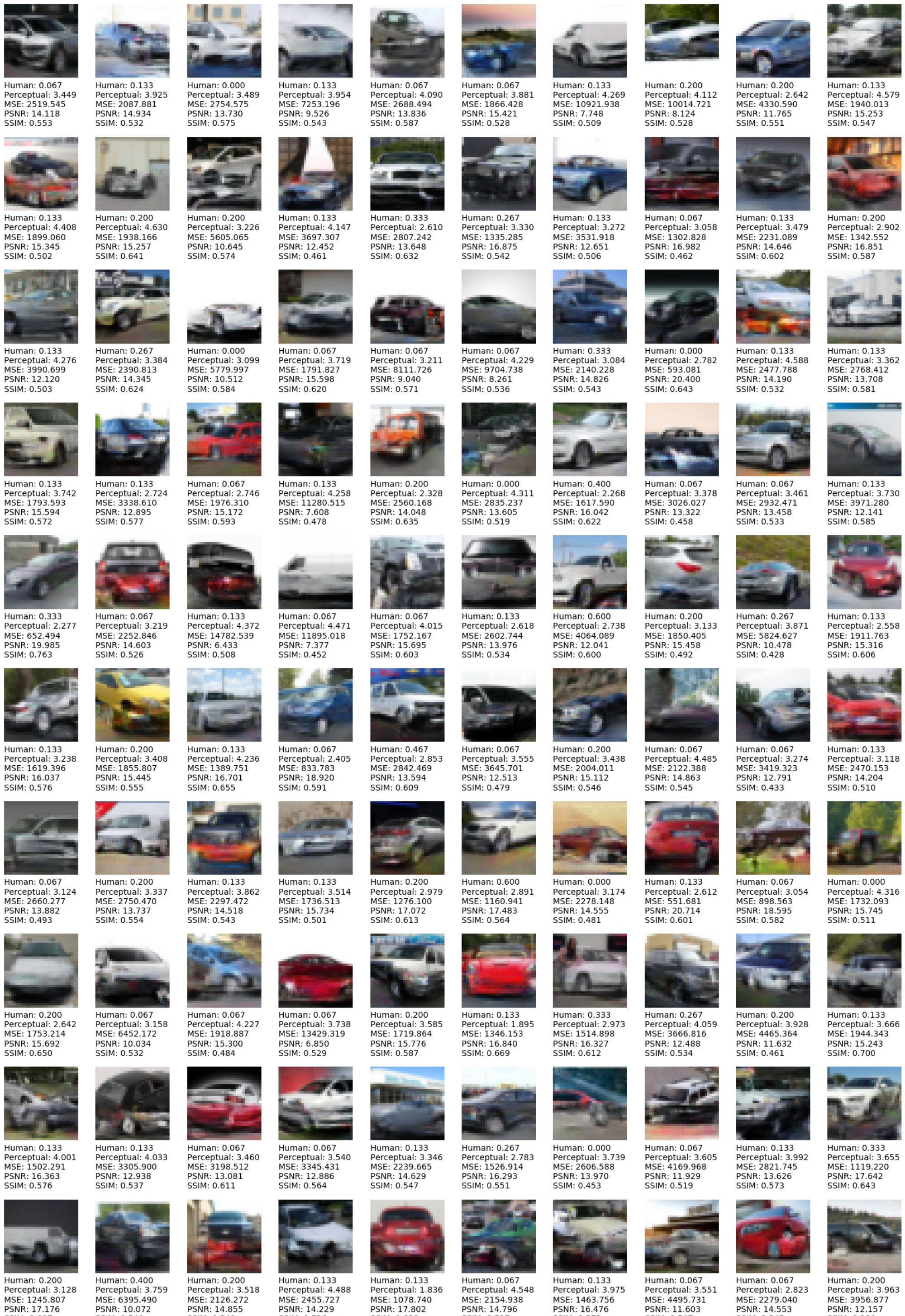

Figure 28: Stanford Cars 32x32 results for WGAN-GP.

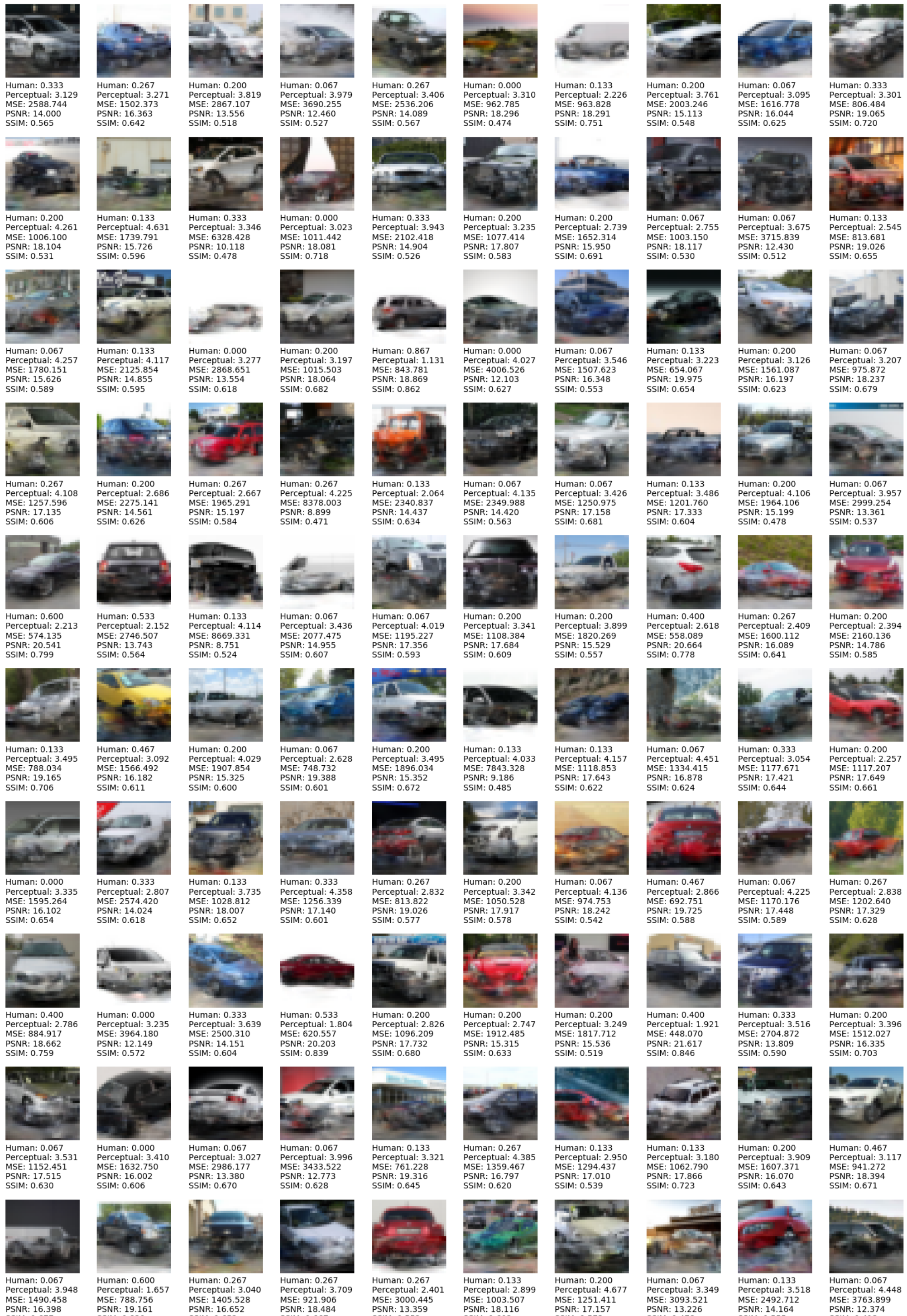

Figure 29: Stanford Cars 32x32 results for Conditional StyleGAN.

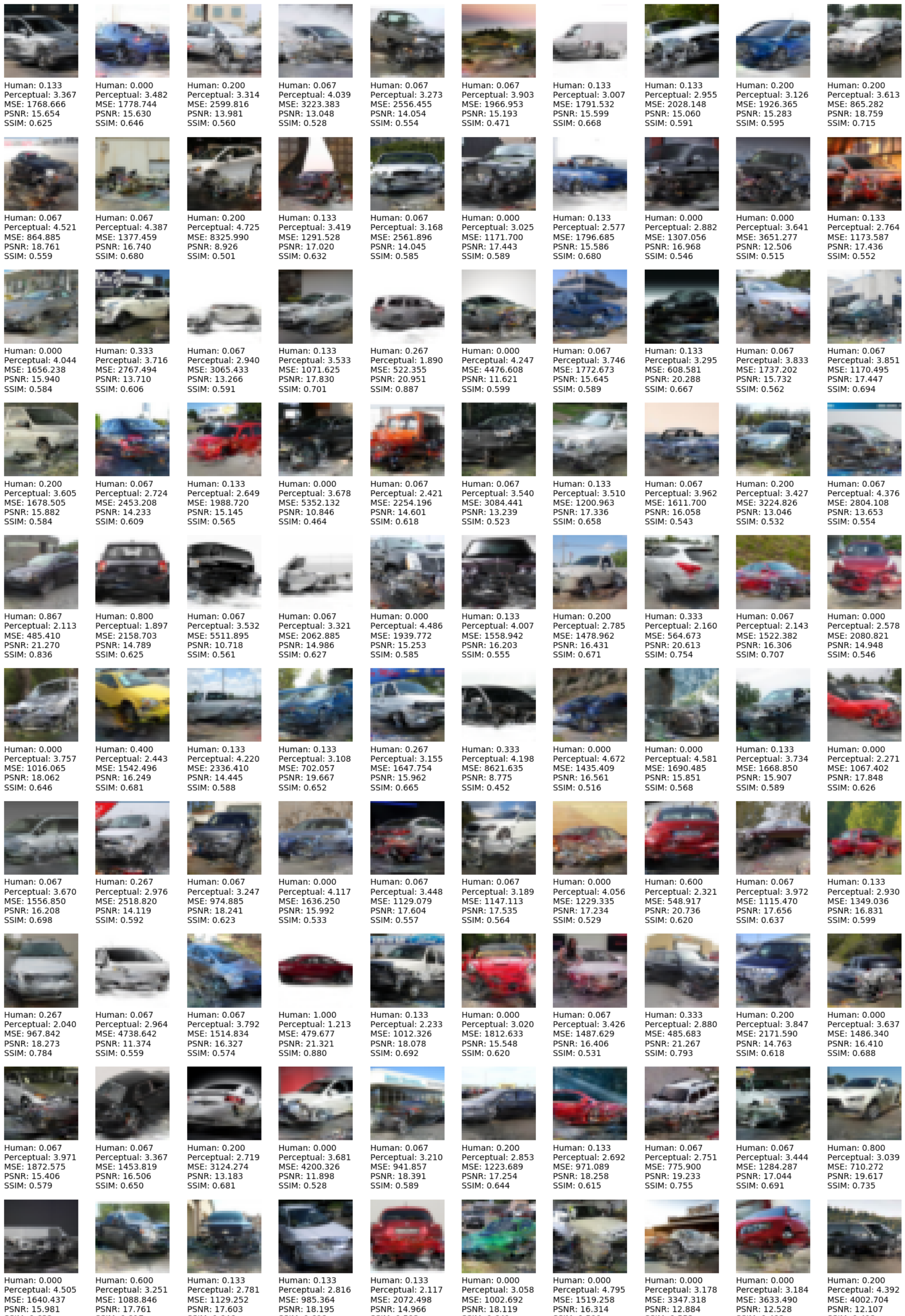

Figure 30: Stanford Cars 32x32 results for Conditional ProGAN.

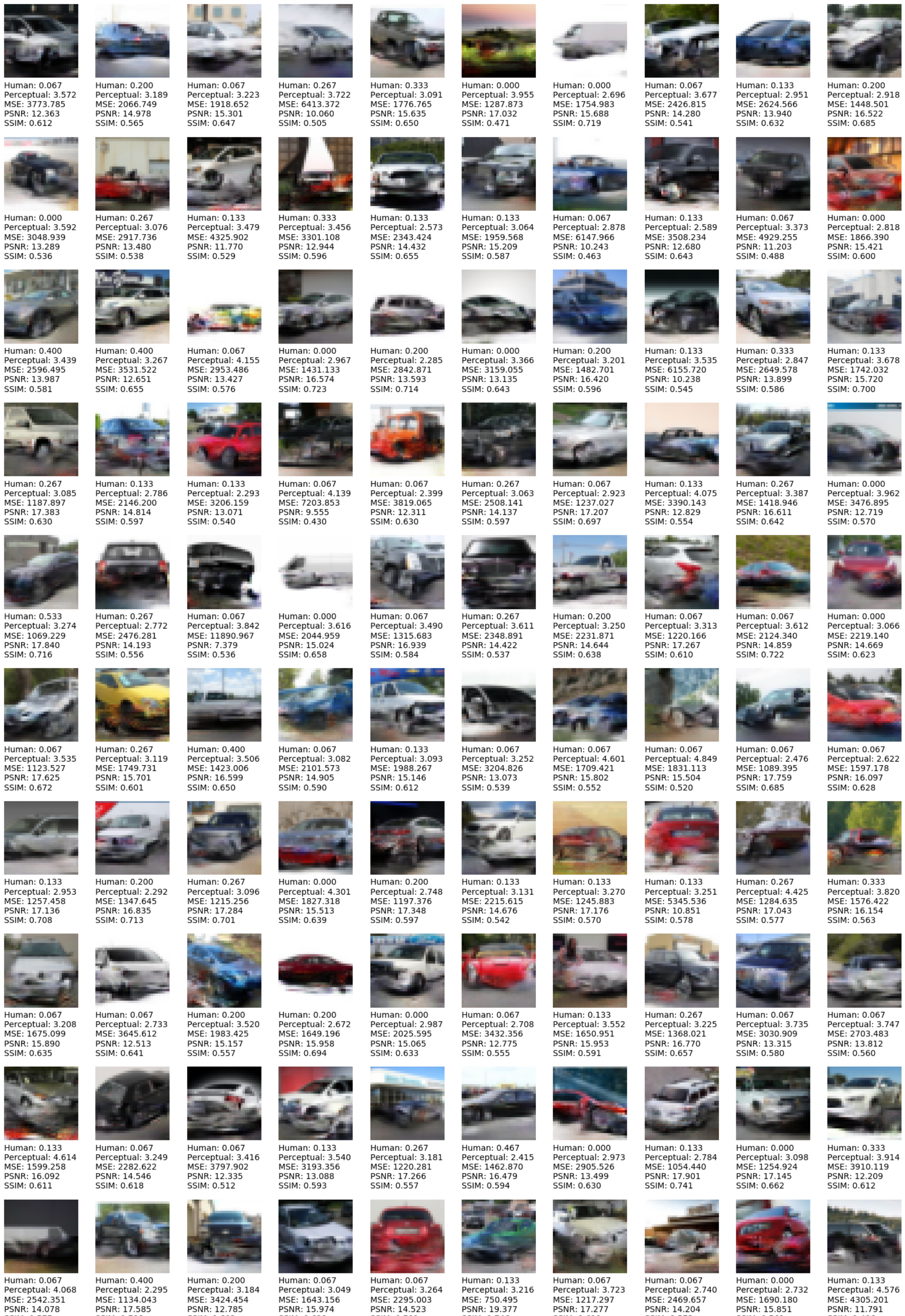

Figure 31: Stanford Cars 32x32 results for Conditional WGAN-GP.

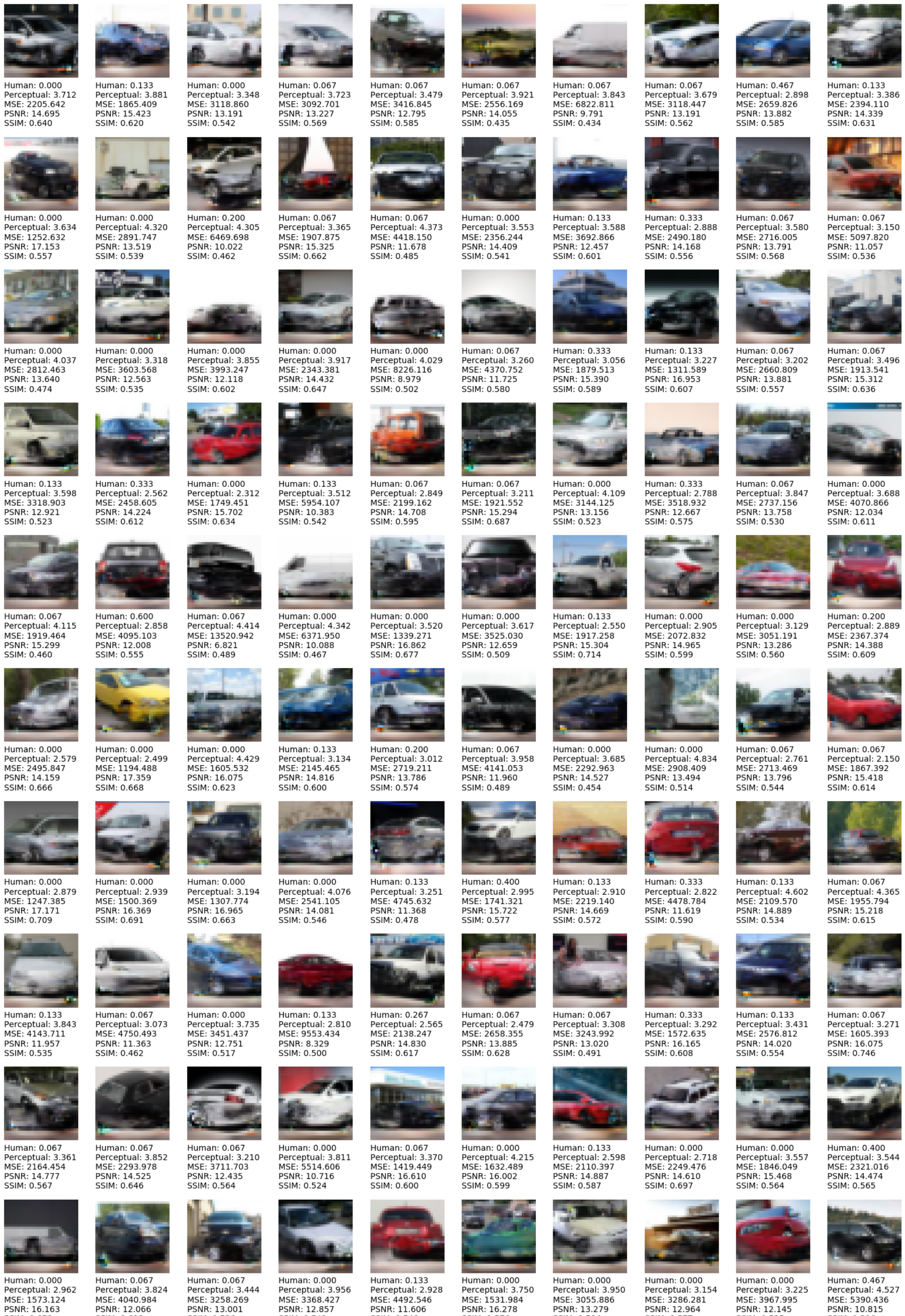

Figure 32: Stanford Cars 32x32 results for DeepFill.

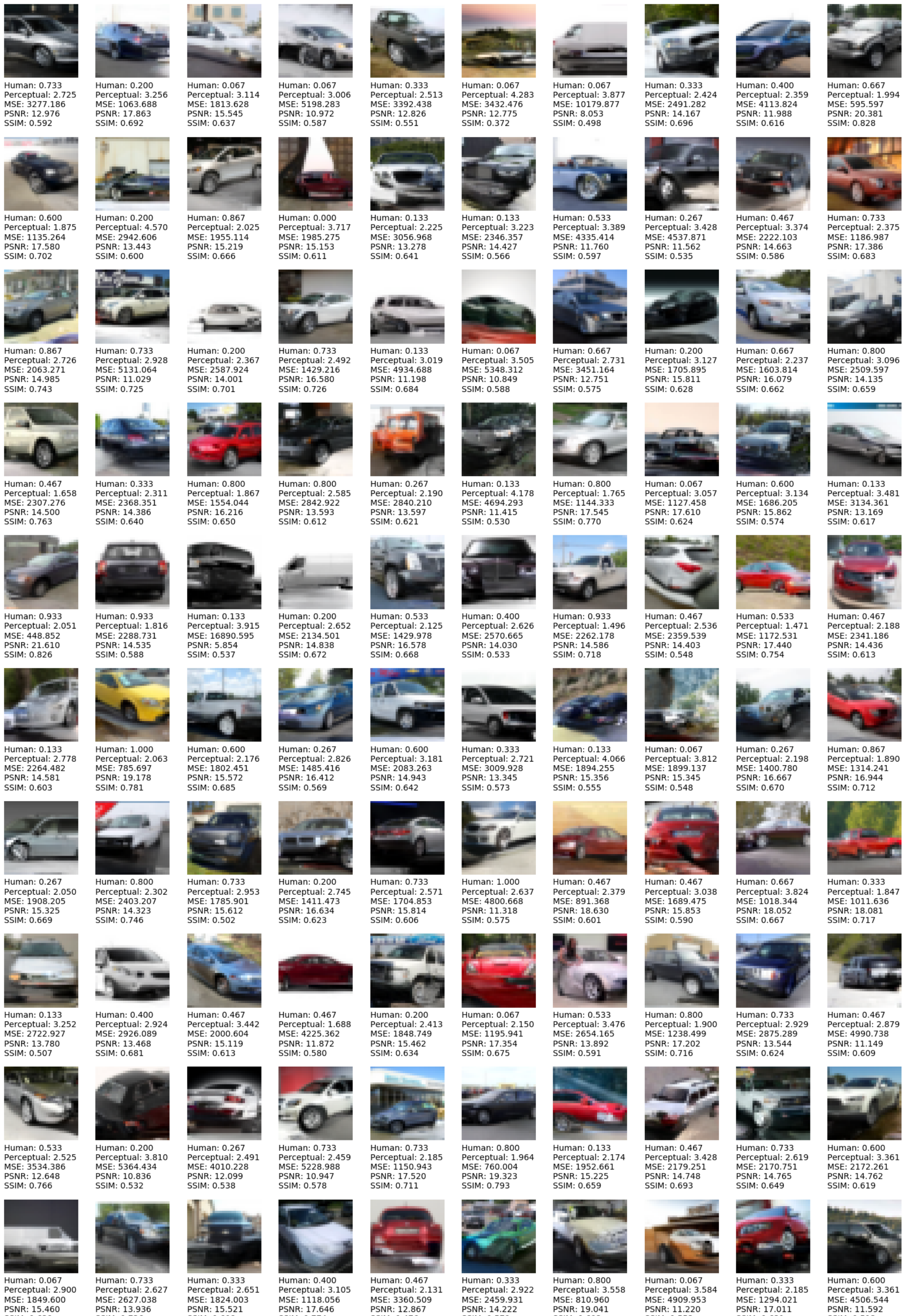

Figure 33: Stanford Cars 32x32 results for PixelCNN++.

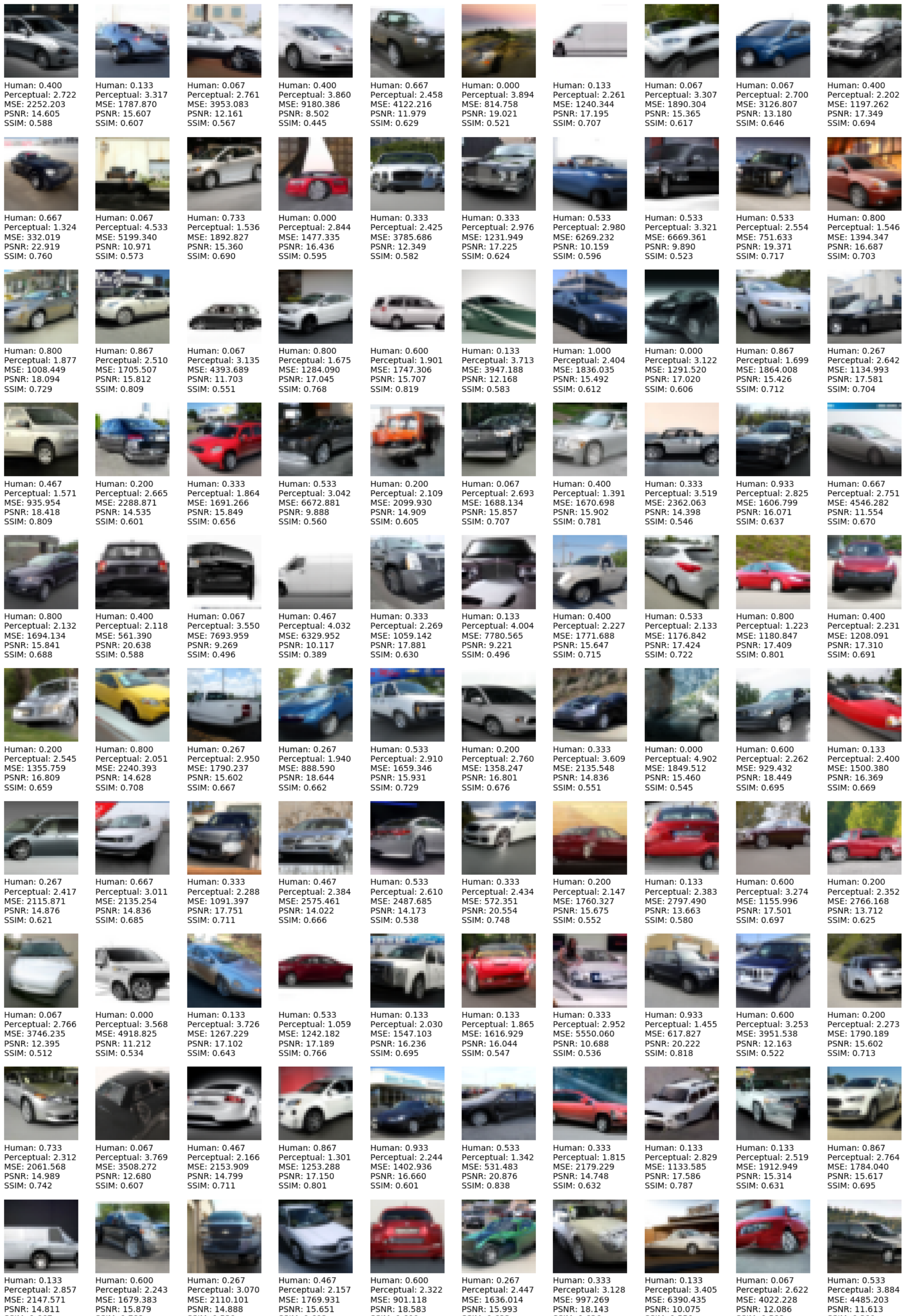

Figure 34: Stanford Cars 32x32 results for PixelSNAIL.

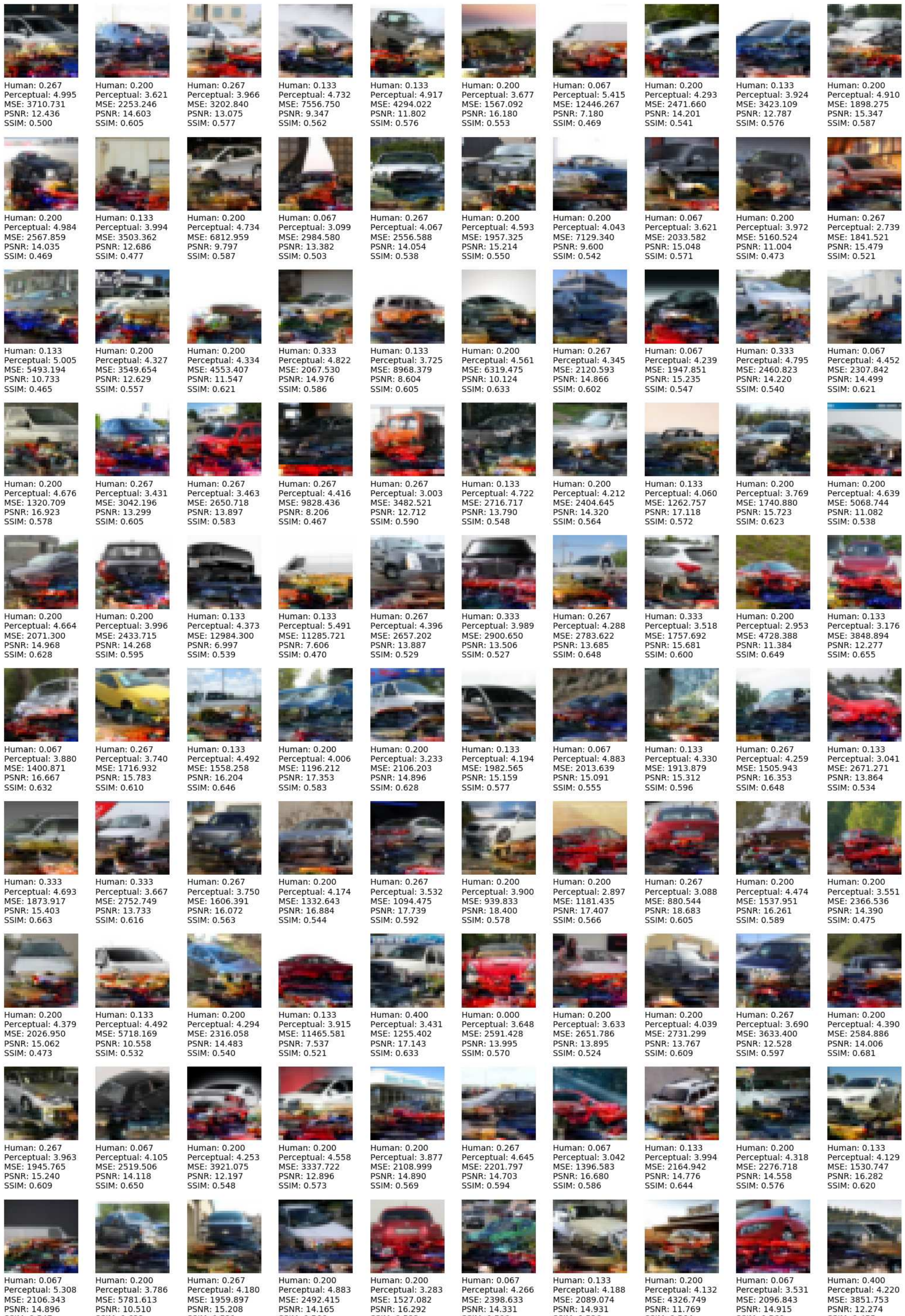

Figure 35: Stanford Cars 32x32 results for Pixel Constrained CNN.

A.2.2 64x64

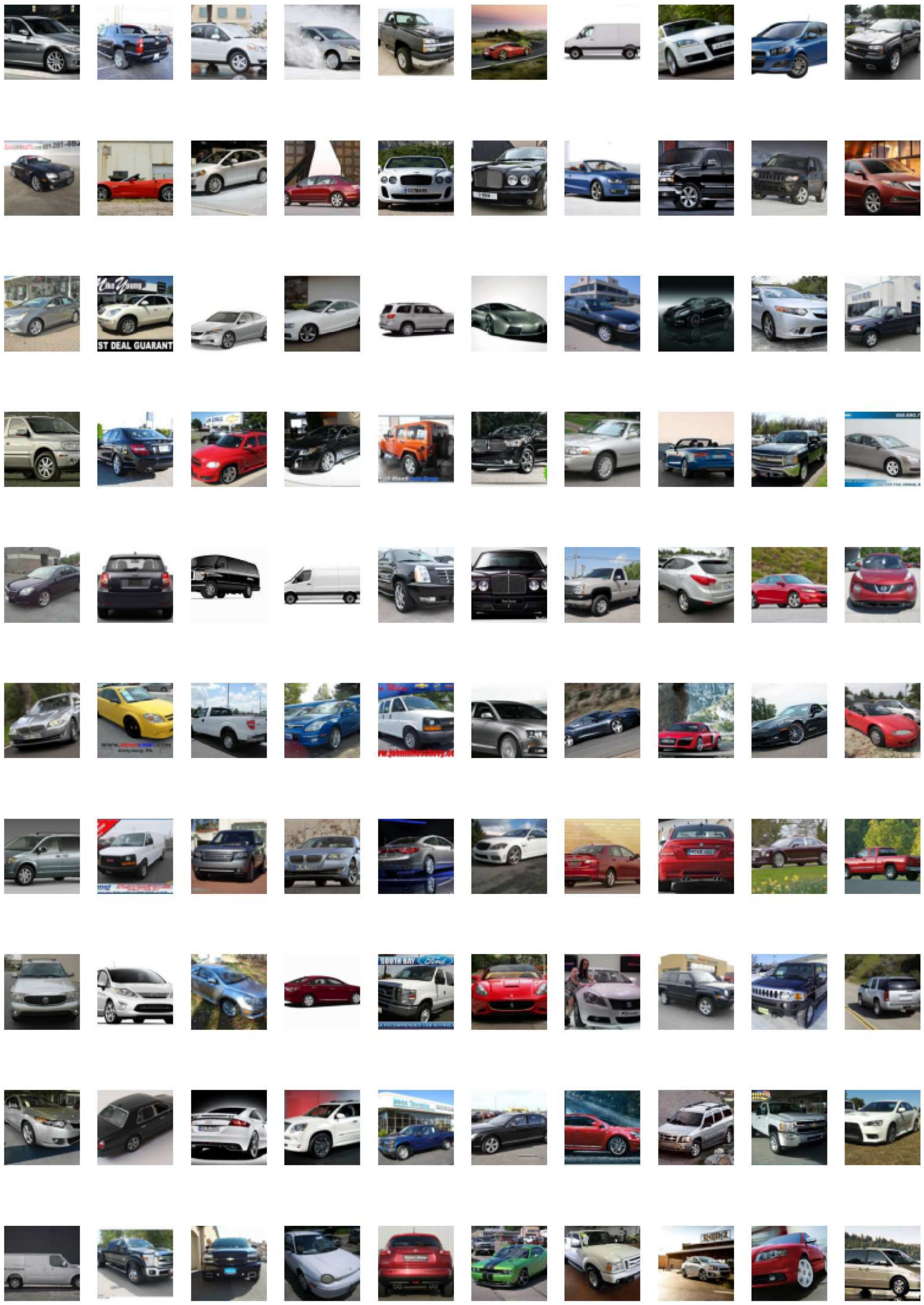

Figure 36: Ground truth for Stanford Cars at 64x64 resolution.

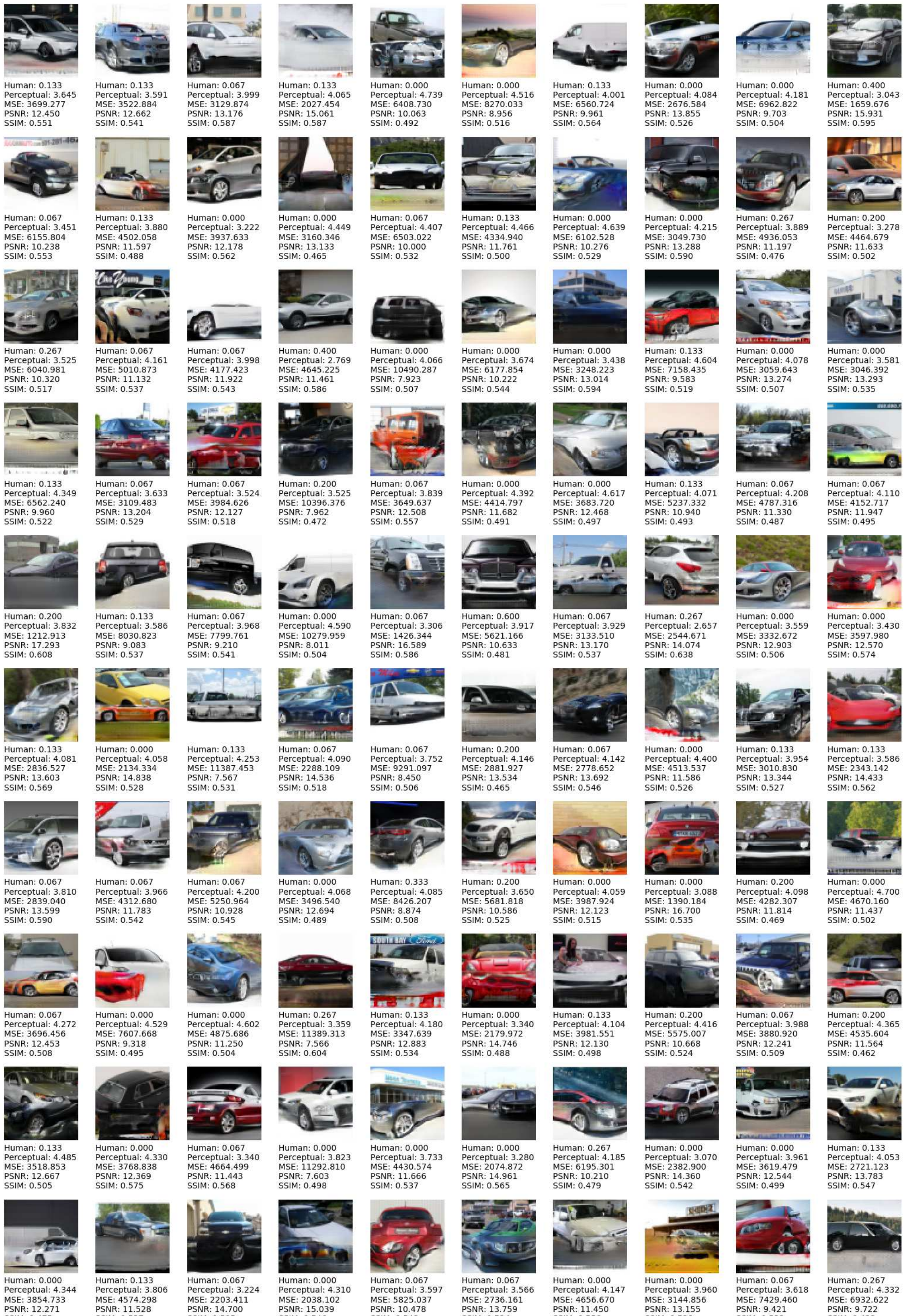

Figure 37: Stanford Cars 64x64 results for ProGAN.

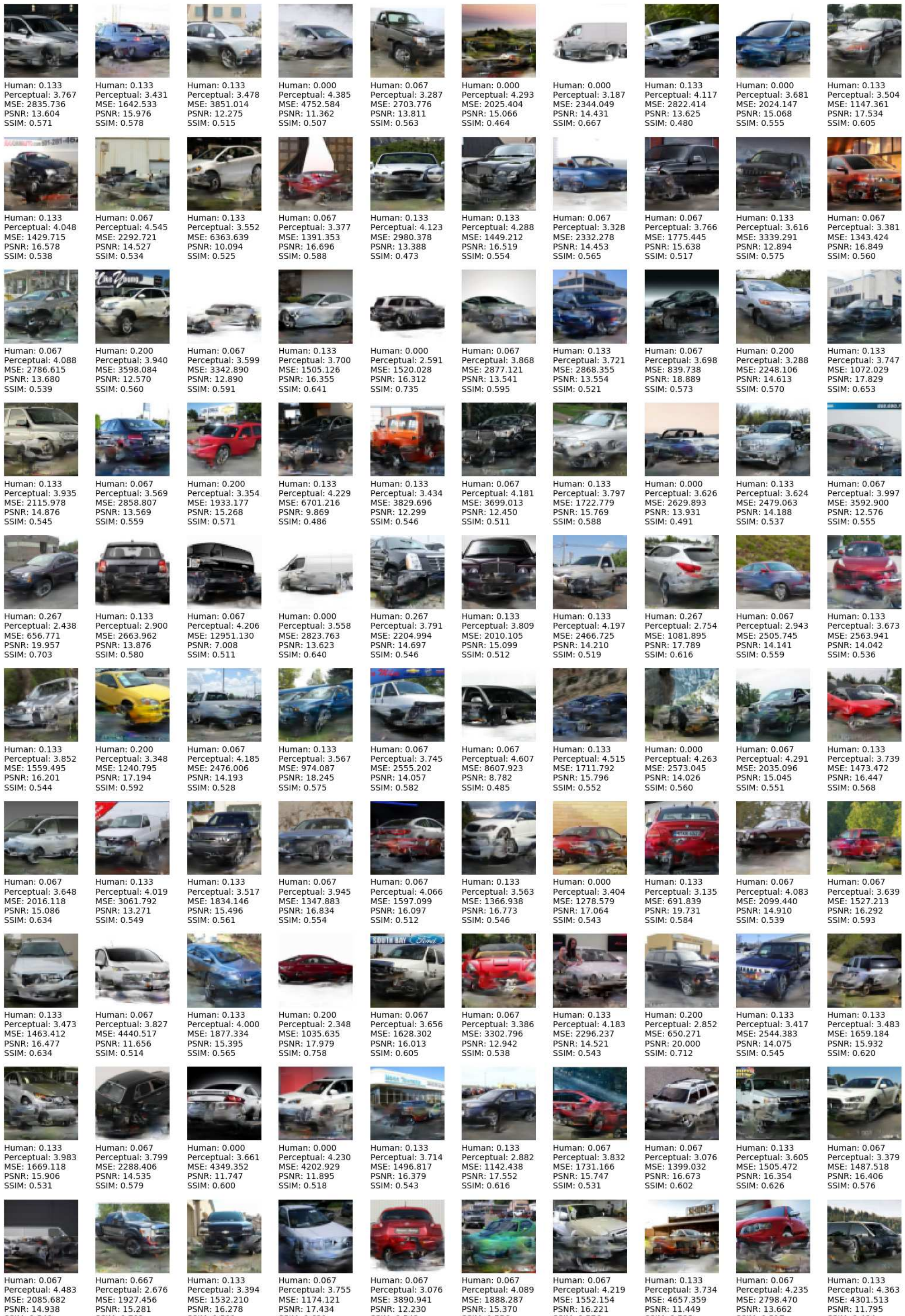

Figure 38: Stanford Cars 64x64 results for Conditional StyleGAN.

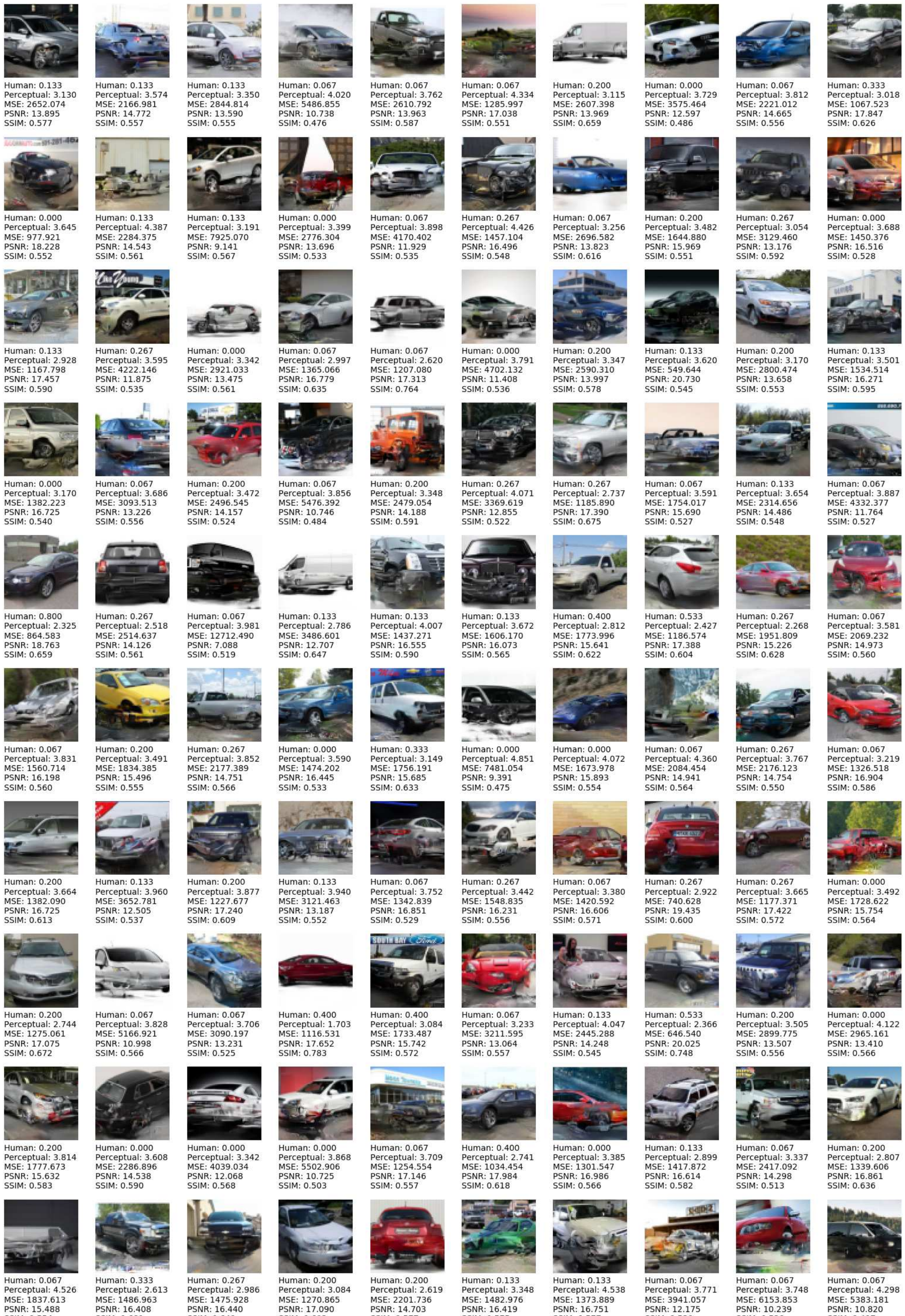

Figure 39: Stanford Cars 64x64 results for Conditional ProGAN.

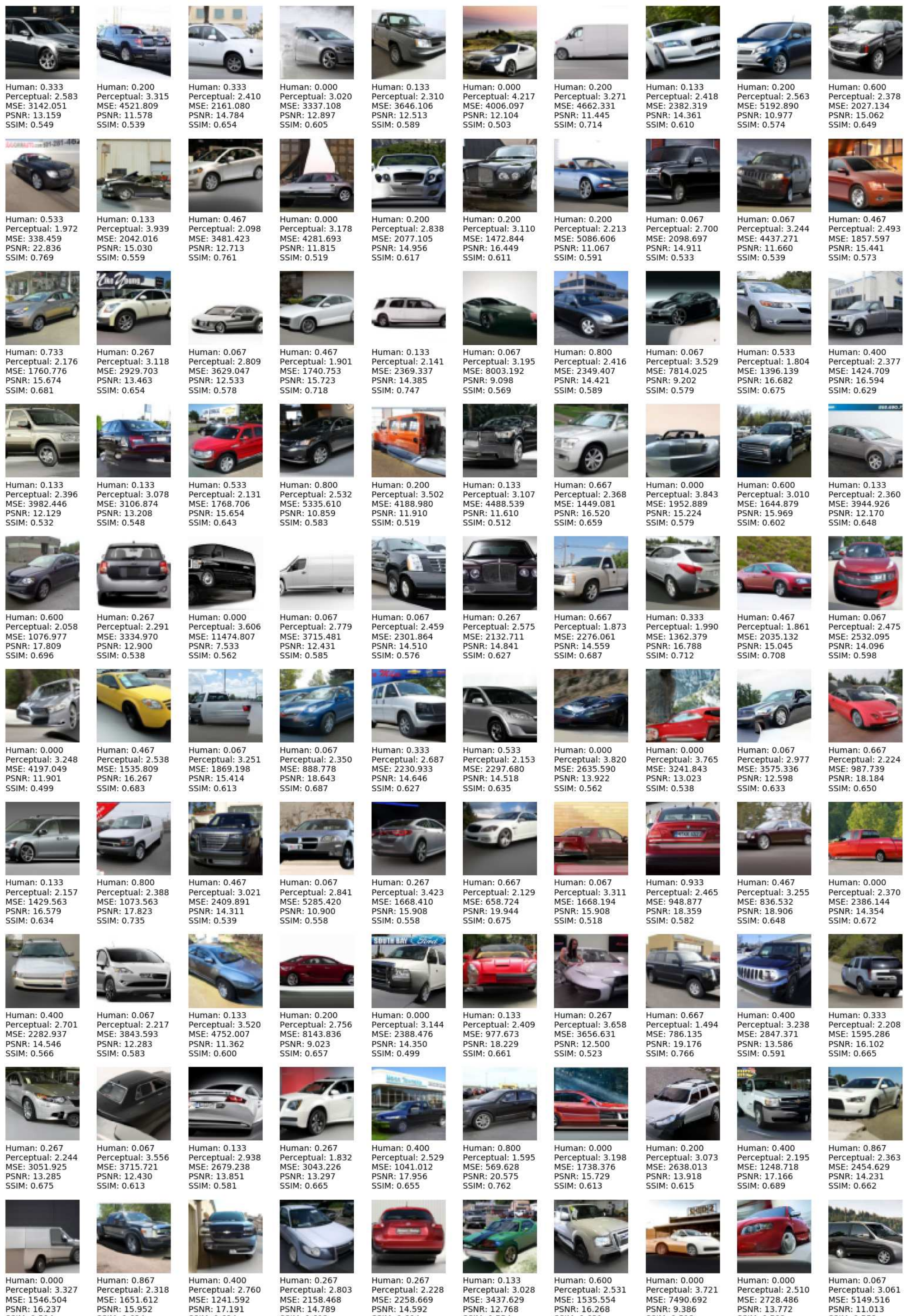

Figure 40: Stanford Cars 64x64 results for PixelCNN++.

A.2.3 128x128

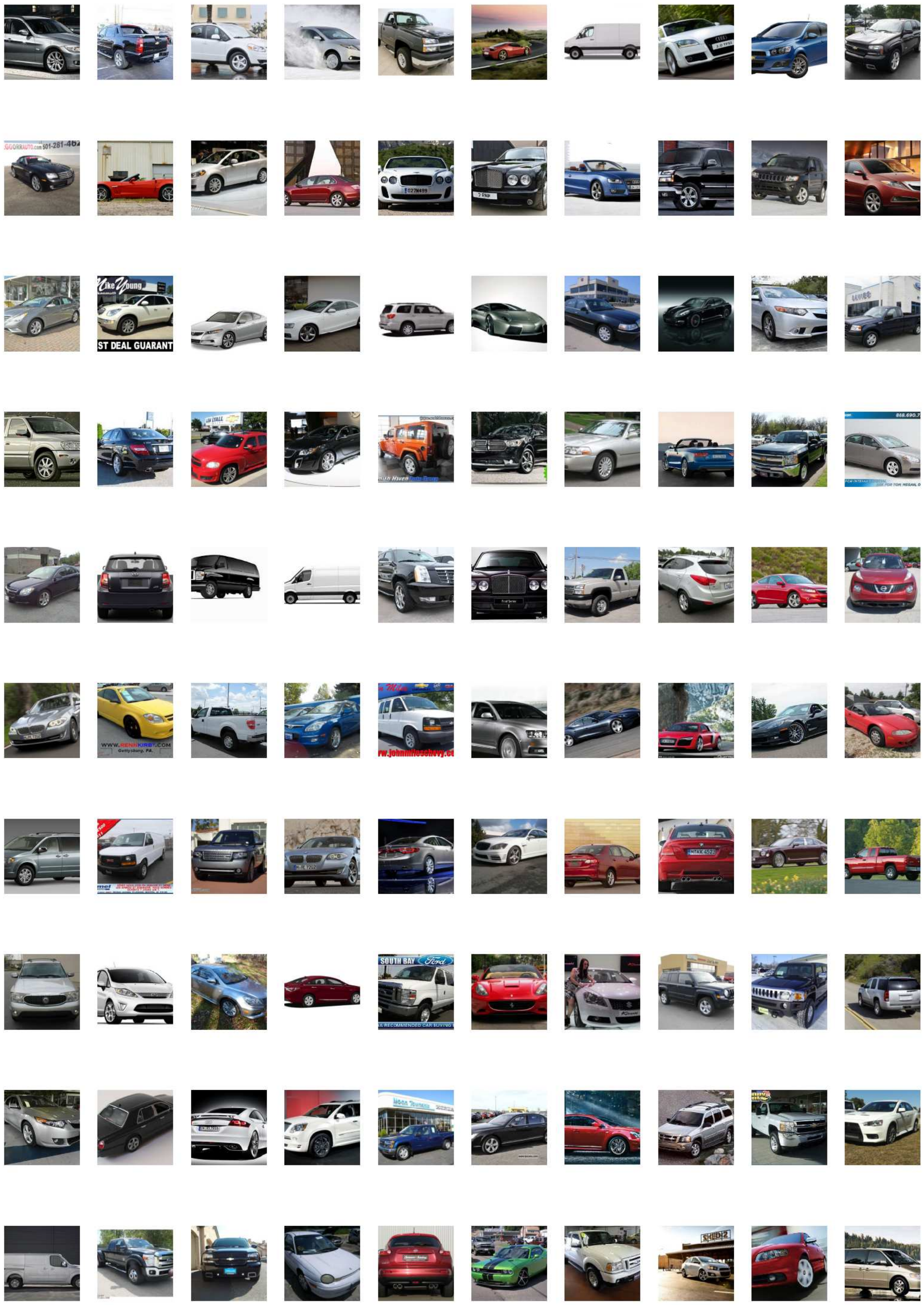

Figure 41: Ground truth for Stanford Cars at 128x128 resolution.

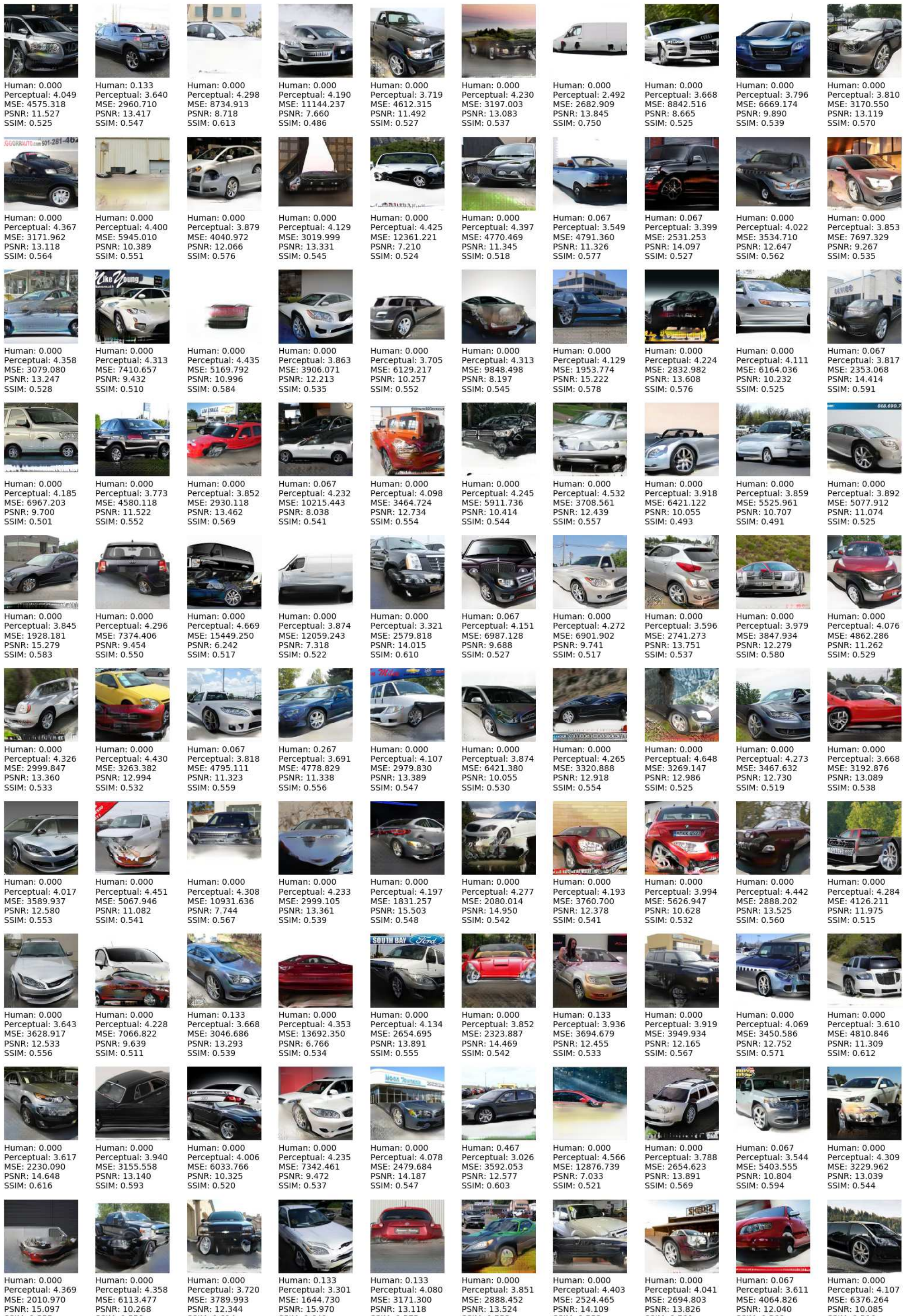

Figure 42: Stanford Cars 128x128 results for ProGAN.

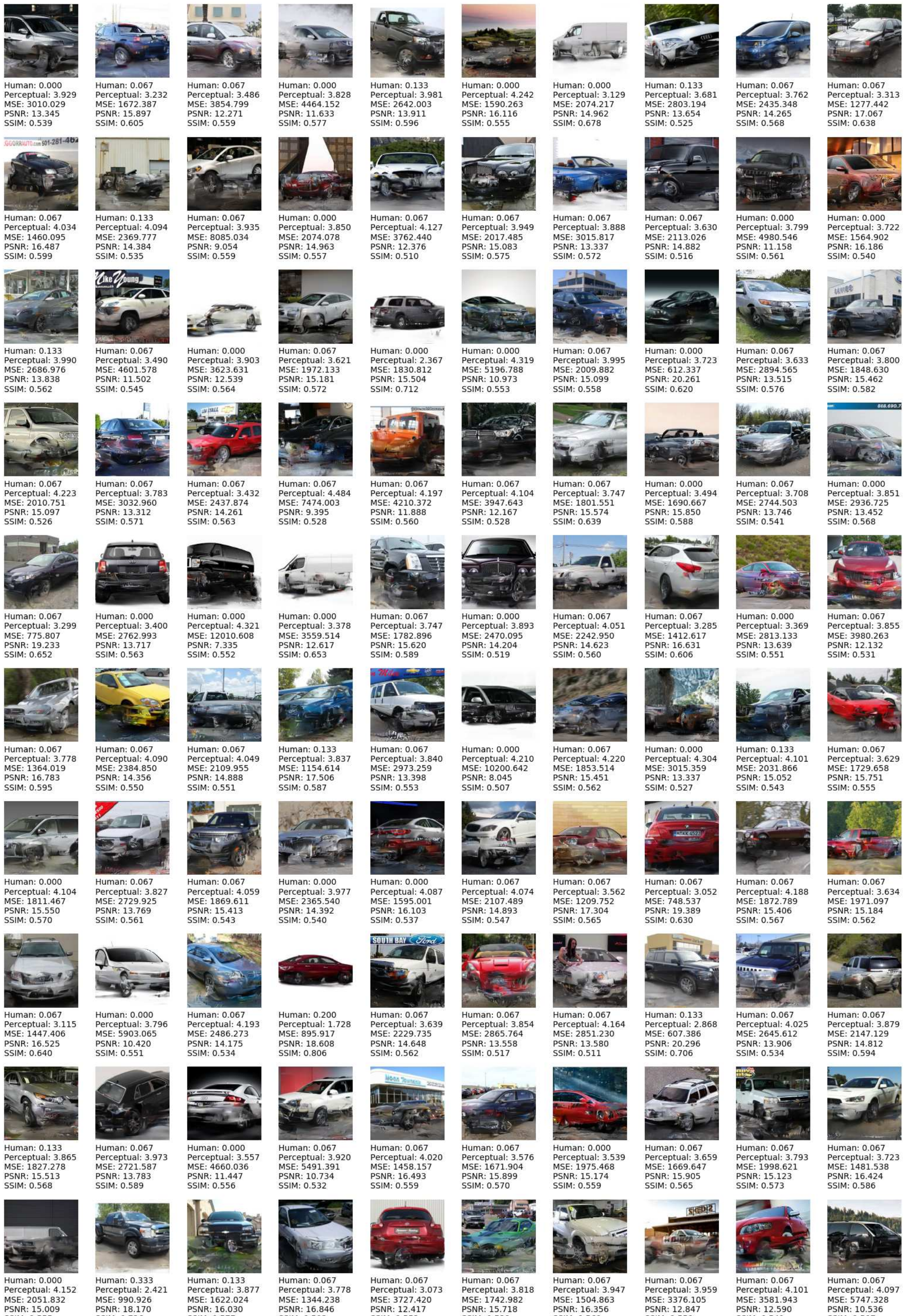

Figure 43: Stanford Cars 128x128 results for Conditional StyleGAN.

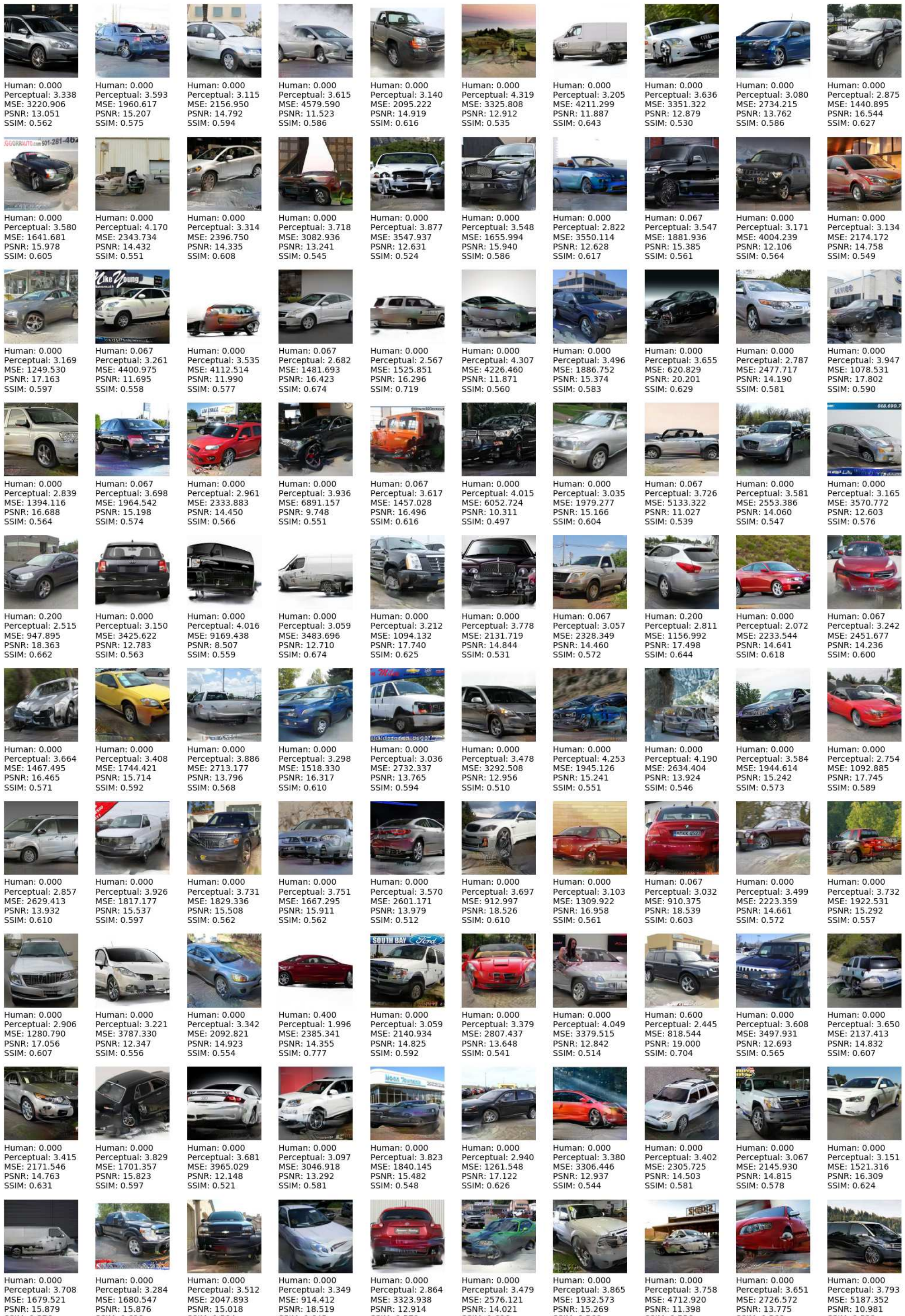

Figure 44: Stanford Cars 128x128 results for Conditional ProGAN.

A.3 CUB

A.3.1 32x32

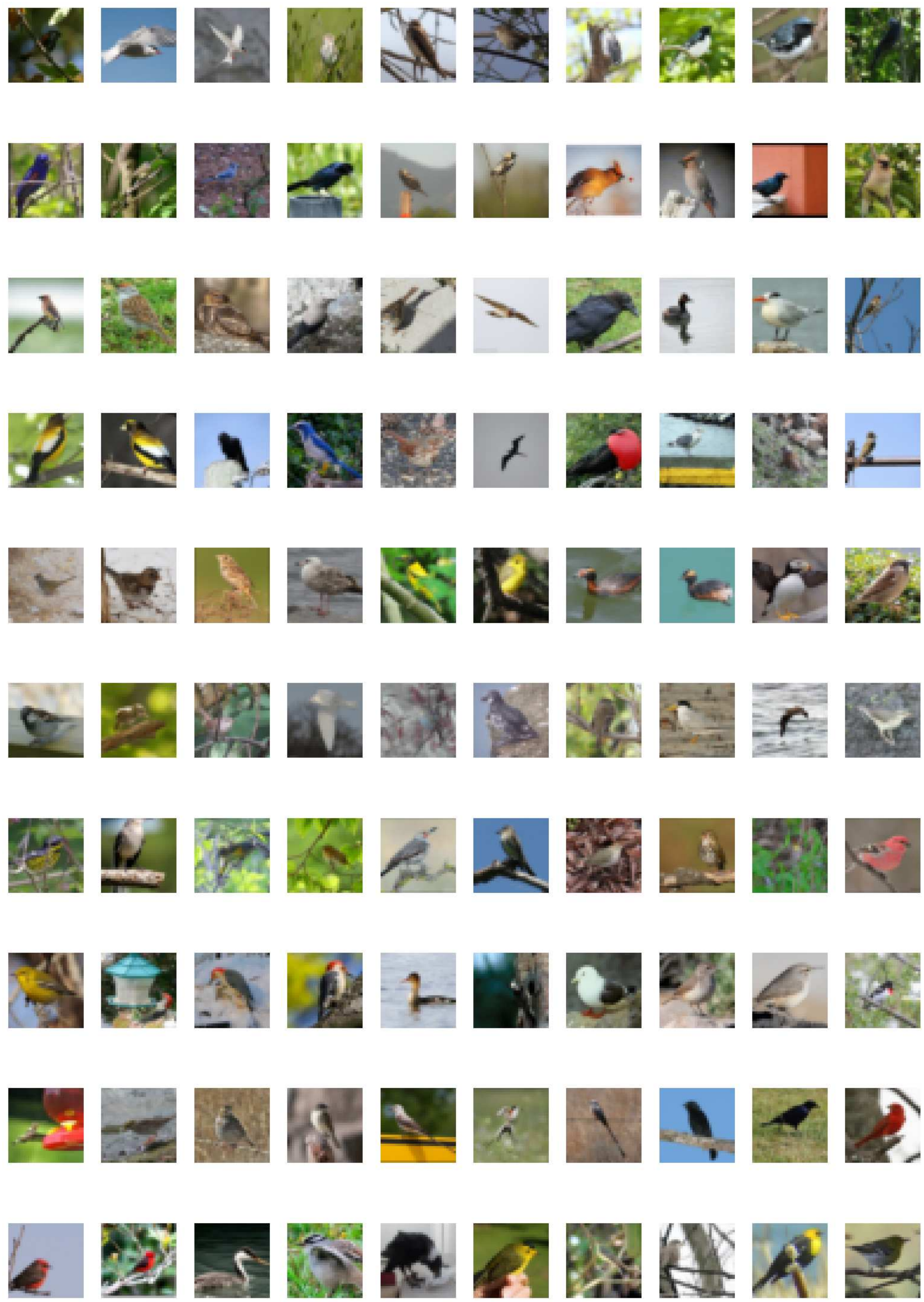

Figure 45: Ground truth for CUB at 32x32 resolution.

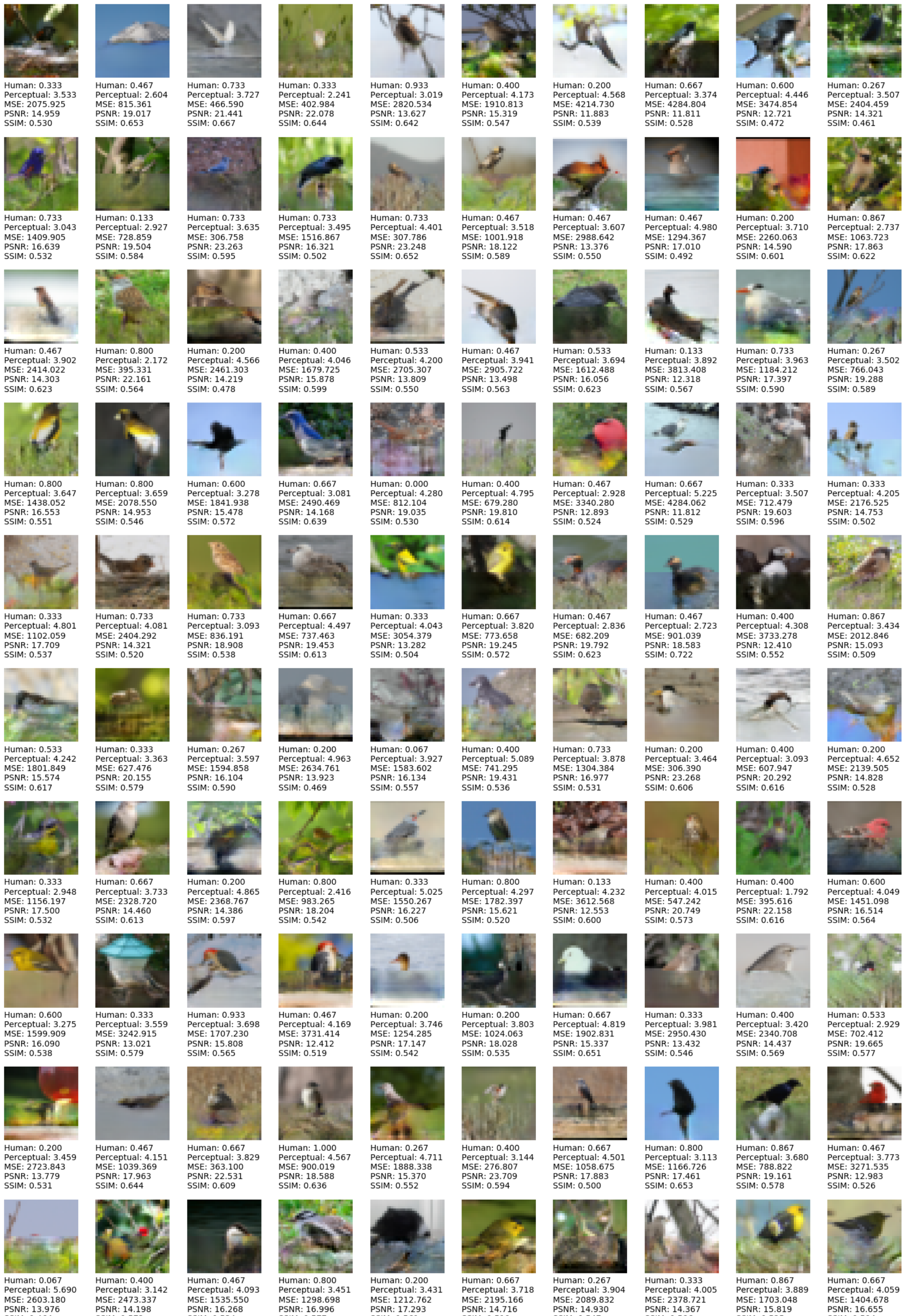

Figure 46: CUB 32x32 results for ProGAN.

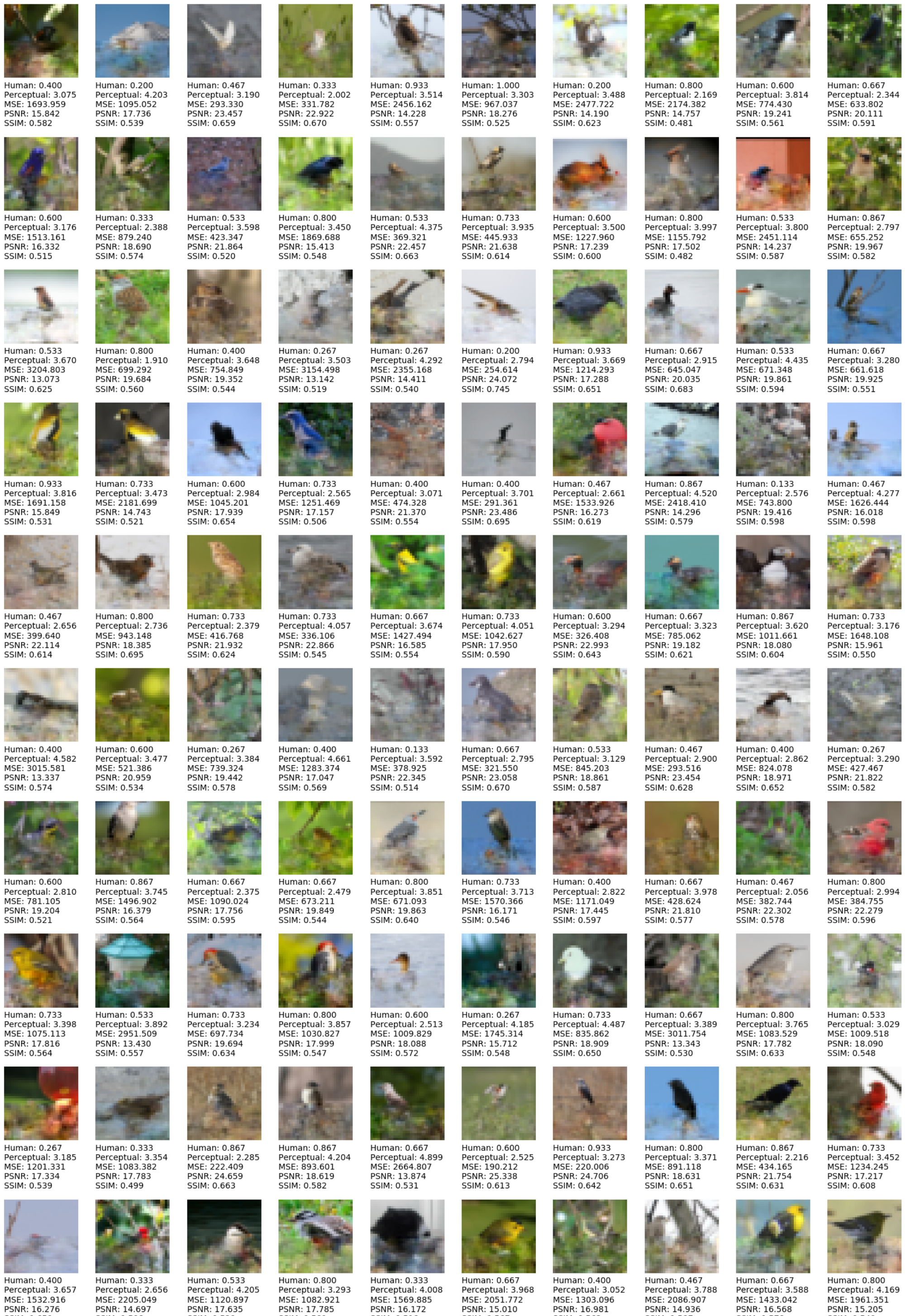

Figure 47: CUB 32x32 results for Conditional StyleGAN.

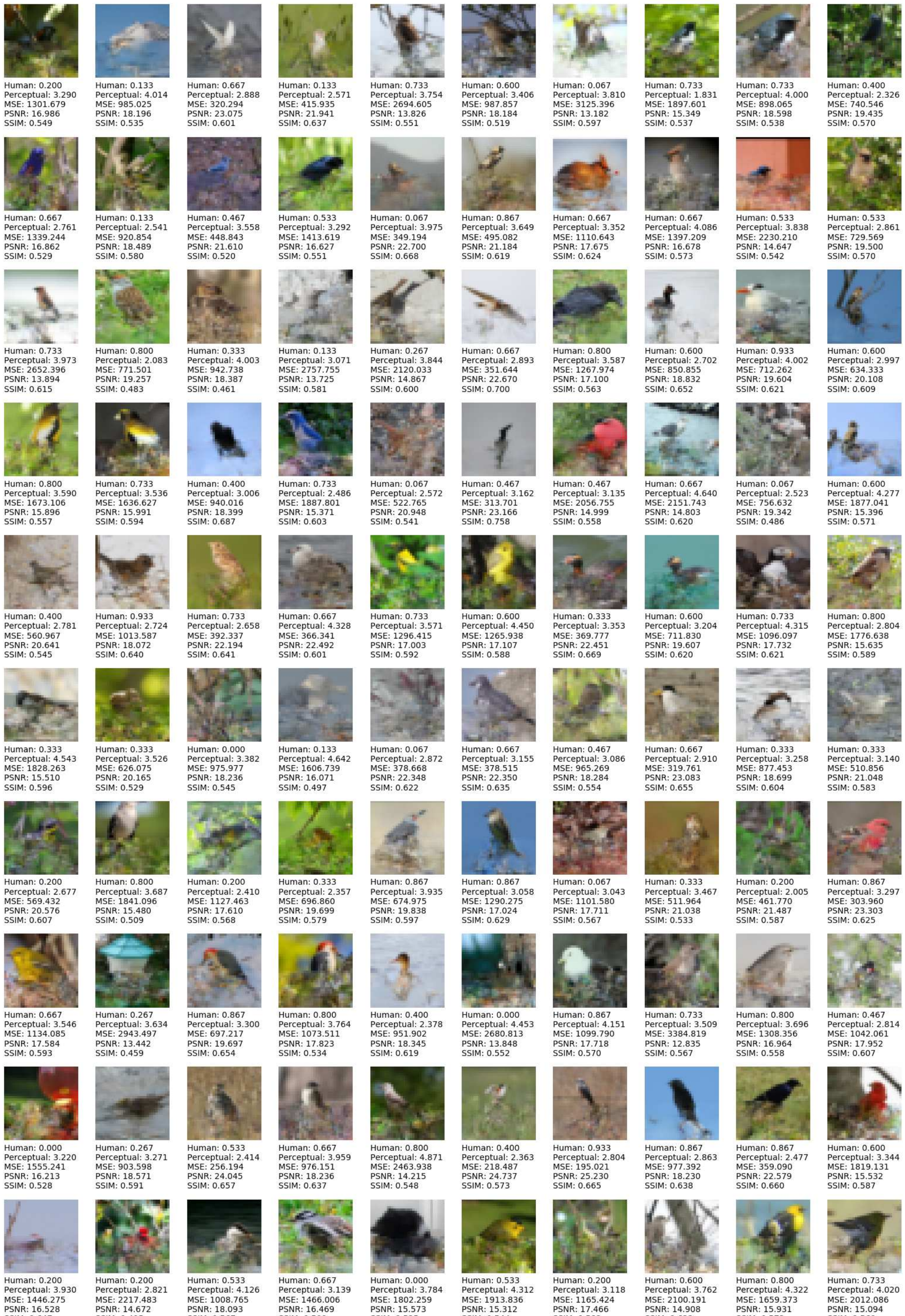

Figure 48: CUB 32x32 results for Conditional ProGAN.

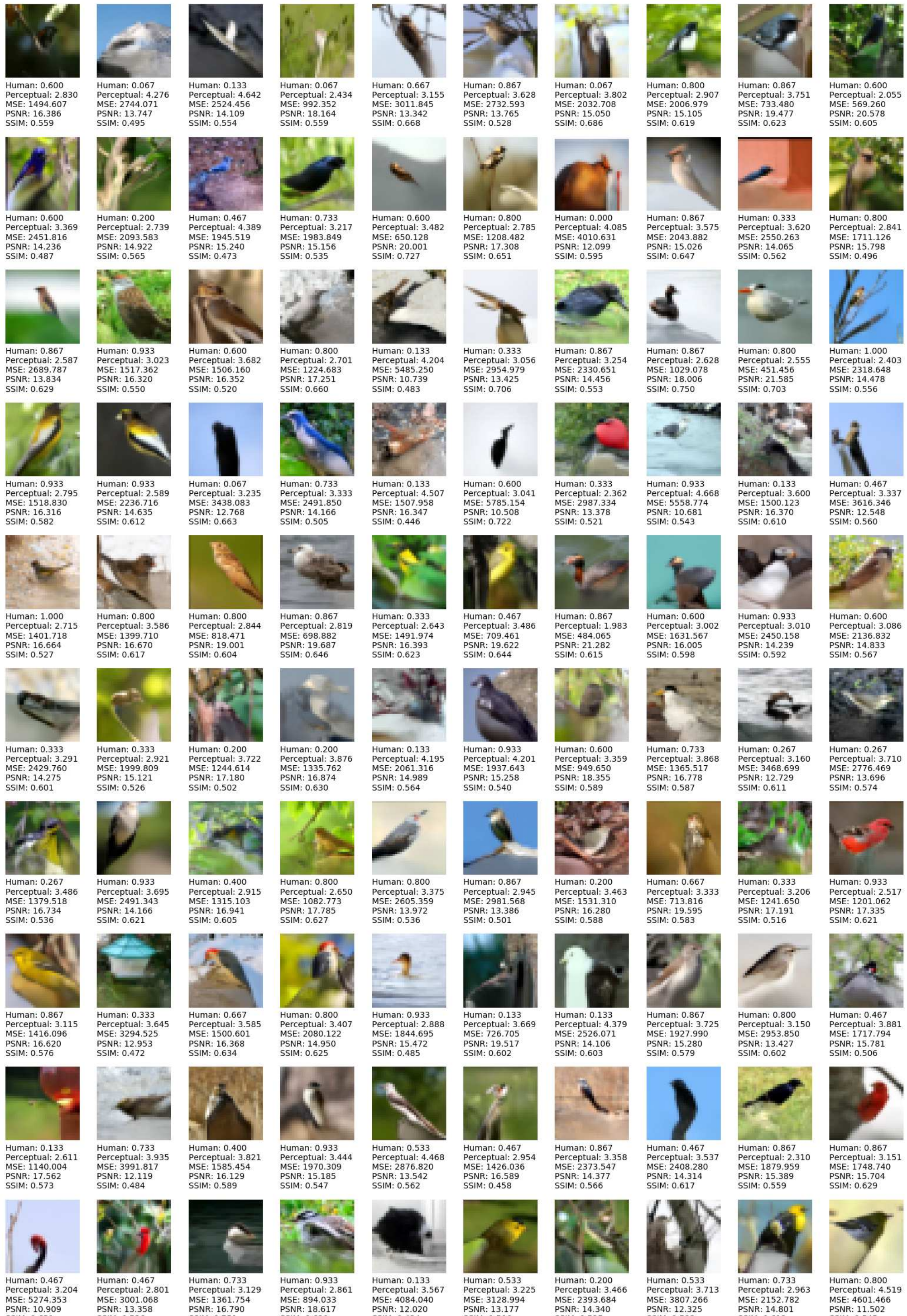

Figure 49: CUB 32x32 results for PixelCNN++.

A.3.2 64x64

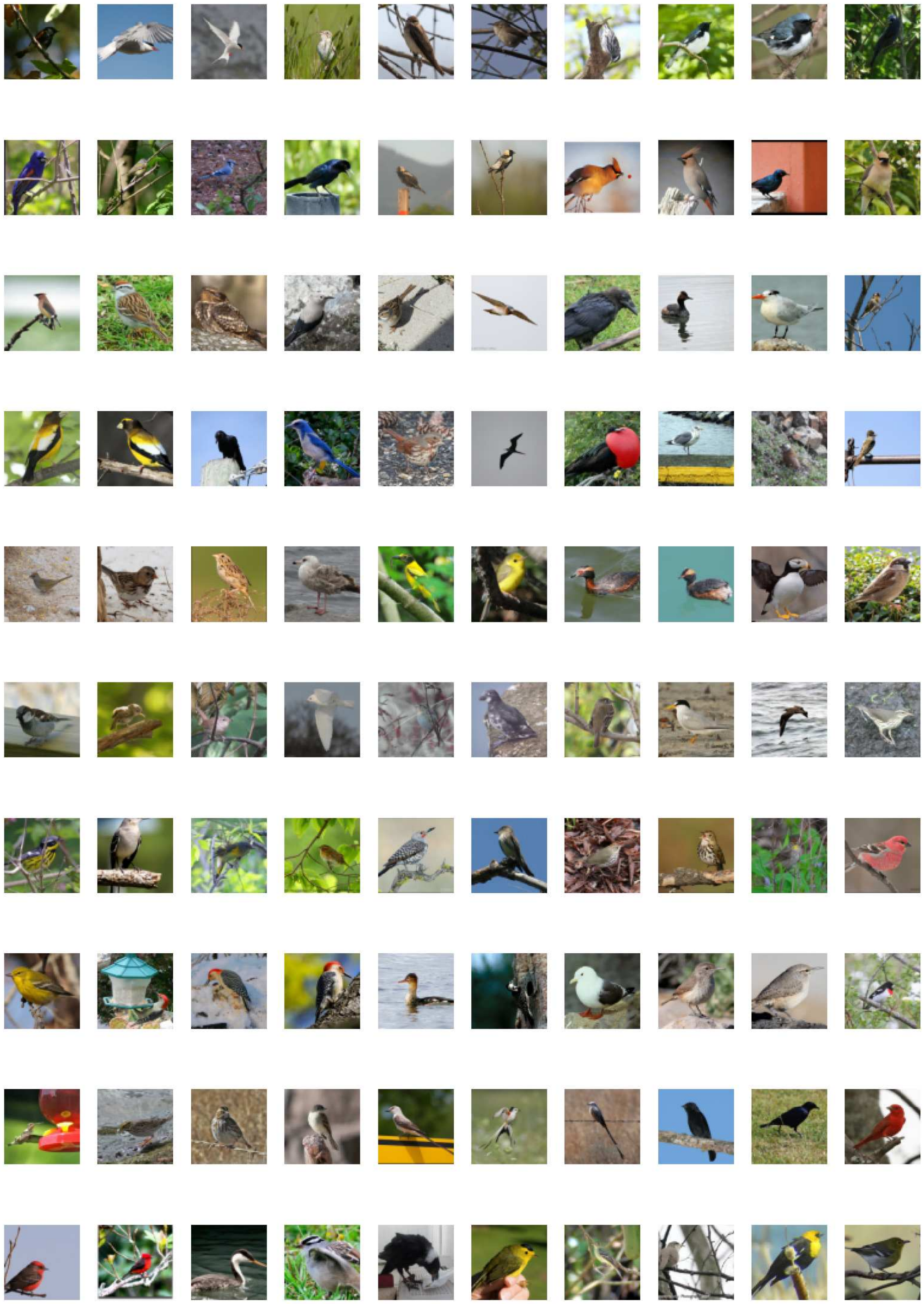

Figure 50: Ground truth for CUB at 64x64 resolution.

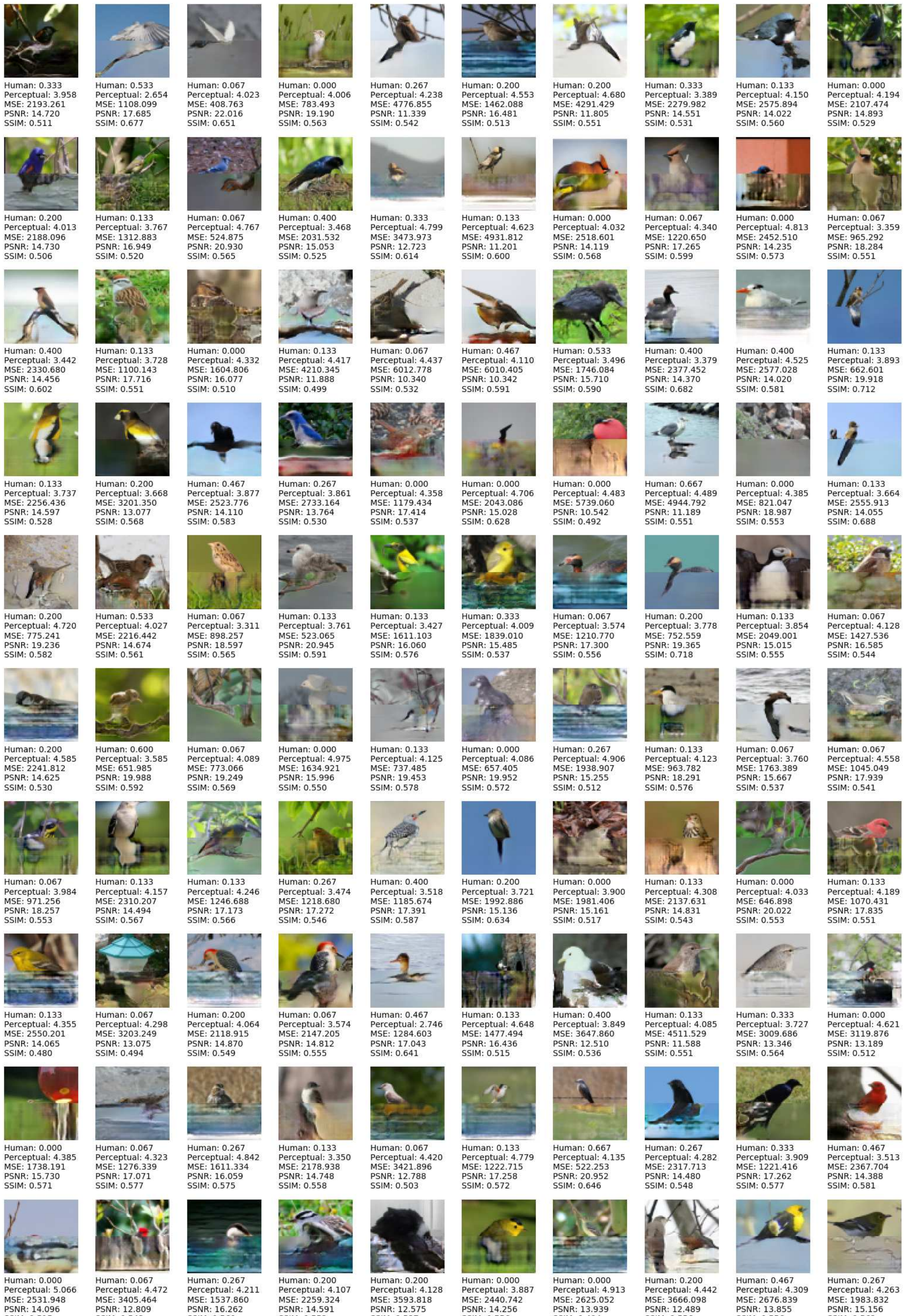

Figure 51: CUB 64x64 results for ProGAN.

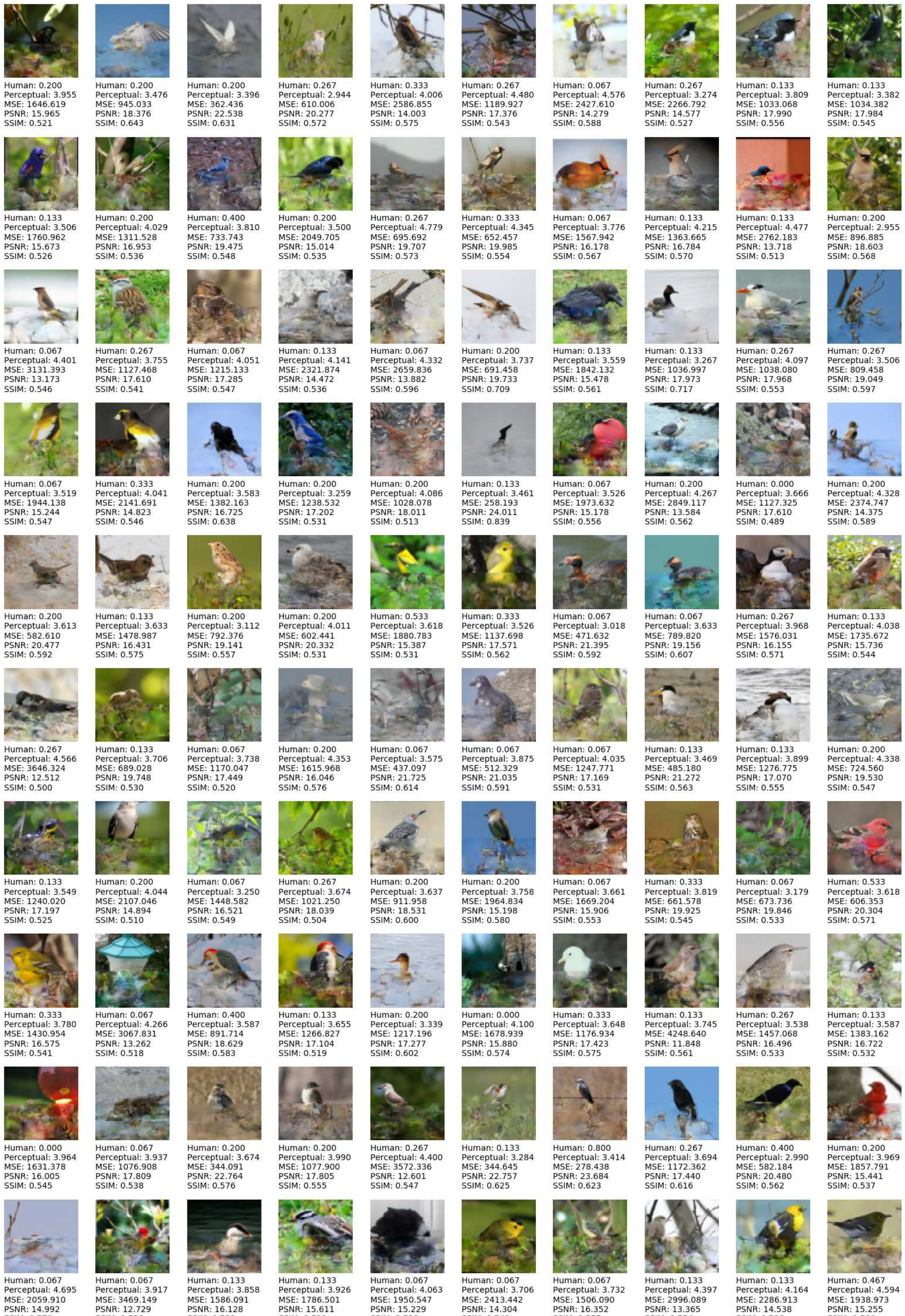

Figure 52: CUB 64x64 results for Conditional StyleGAN.

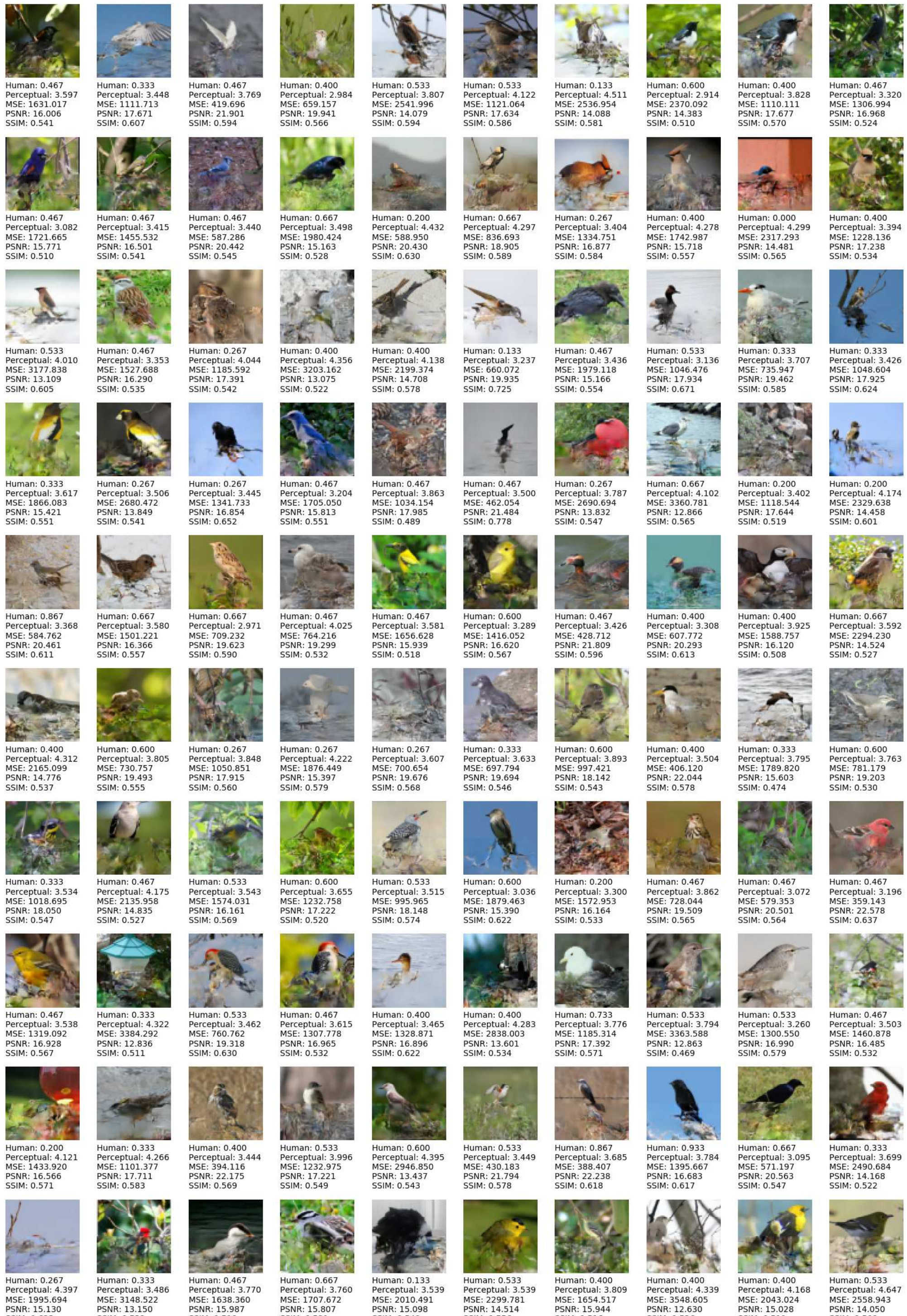

Figure 53: CUB 64x64 results for Conditional ProGAN.

Figure 54: CUB 64x64 results for PixelCNN++.

A.3.3 128x128

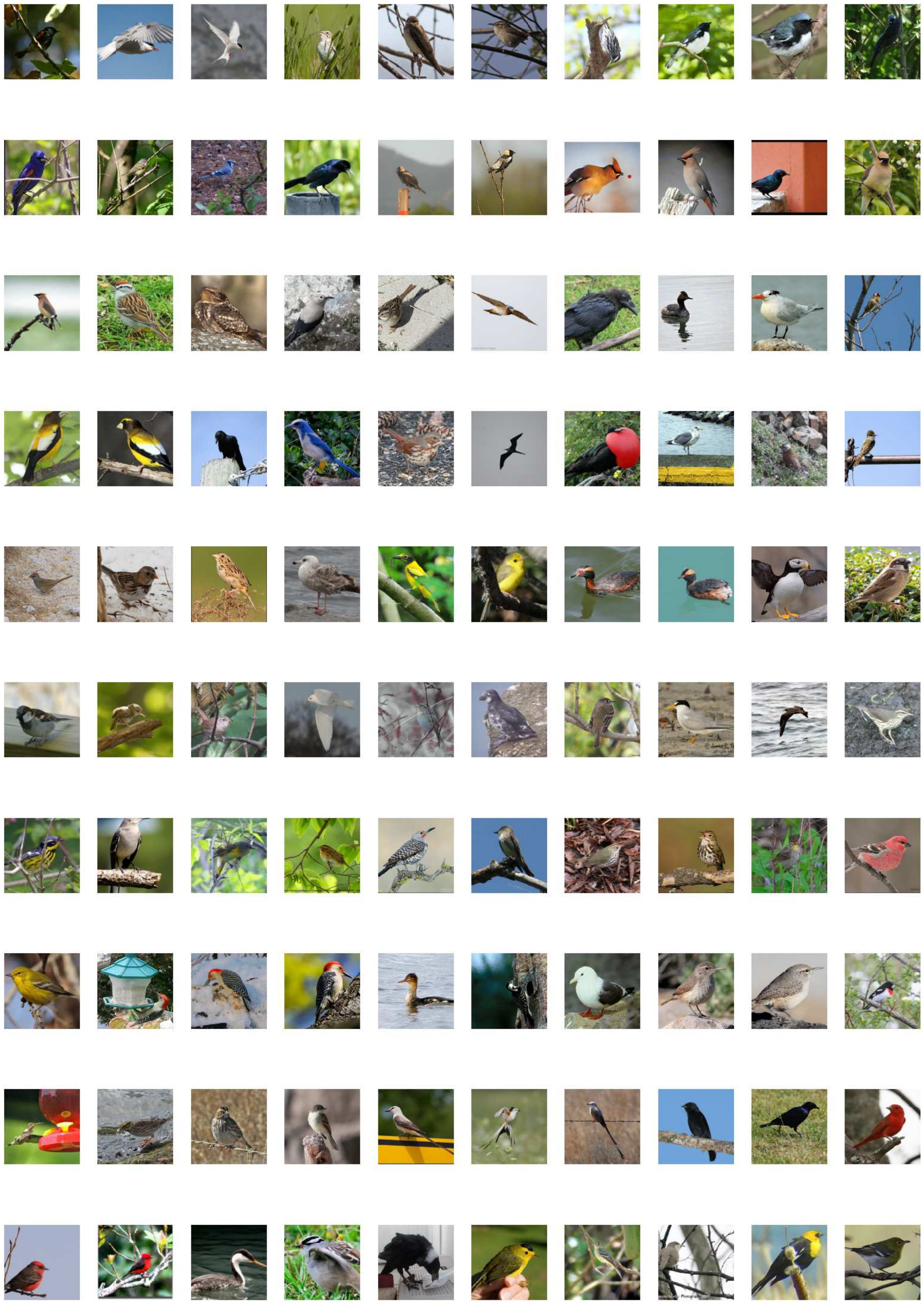

Figure 55: Ground truth for CUB at 128x128 resolution.

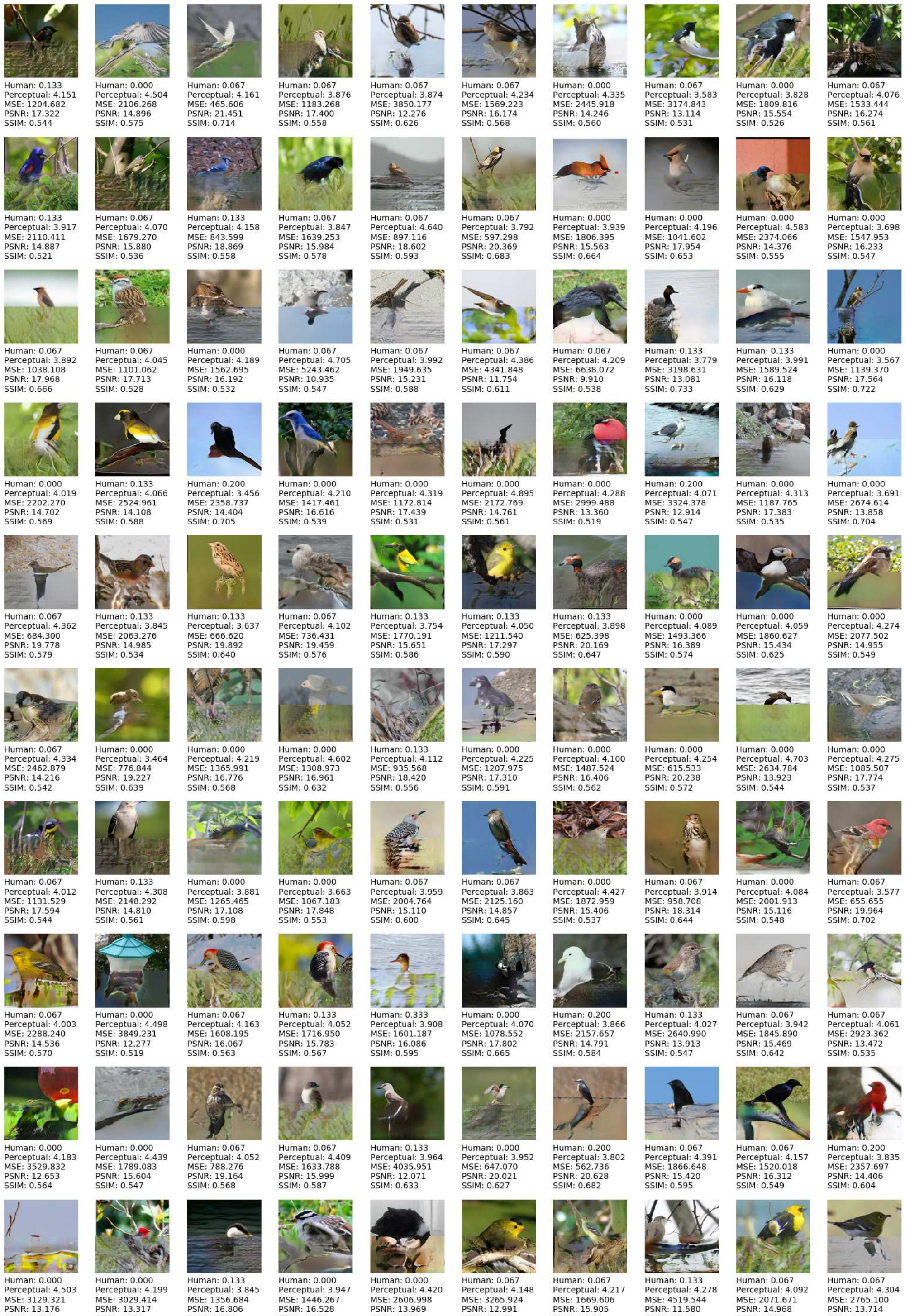

Figure 56: CUB 128x128 results for ProGAN.

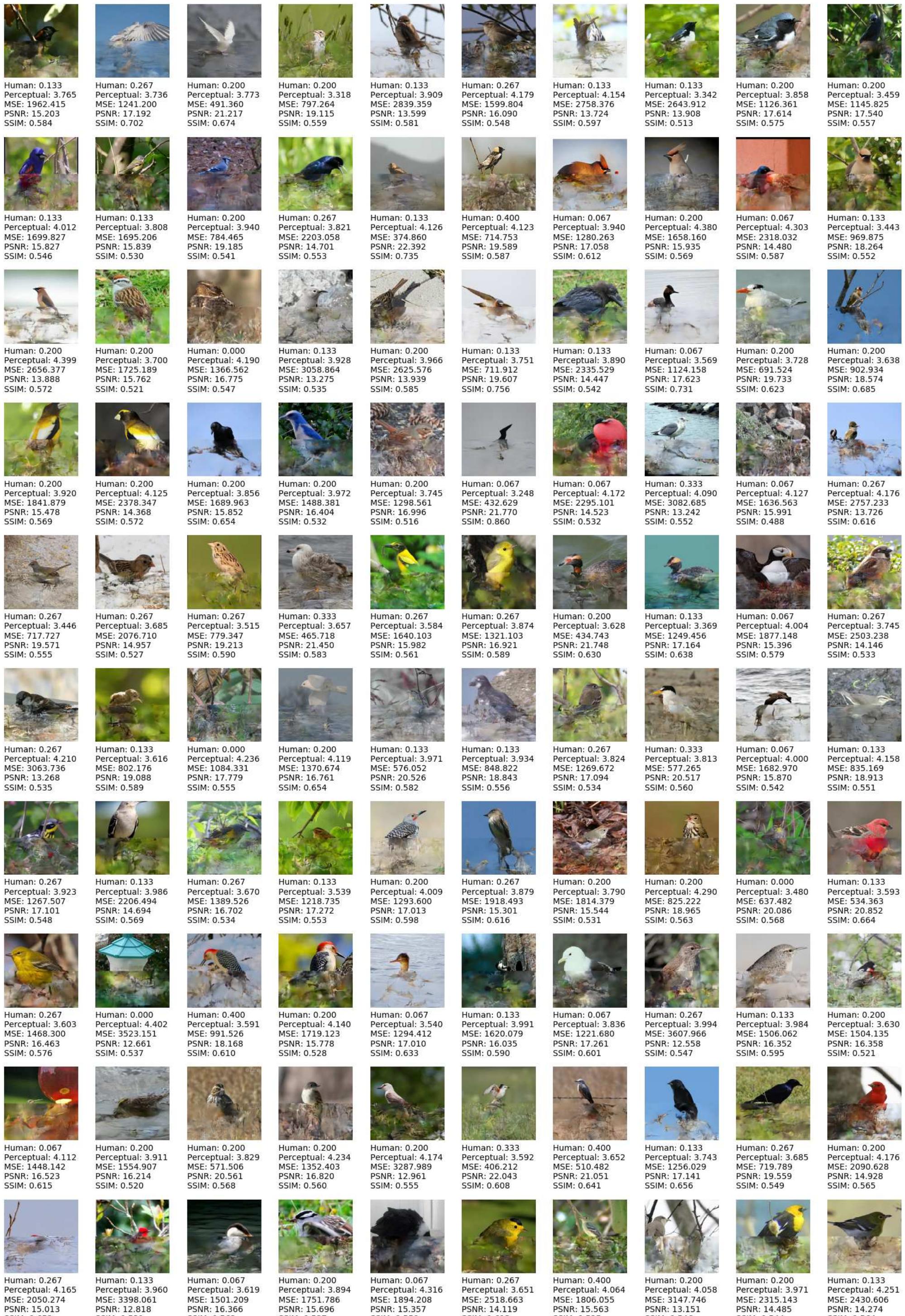

Figure 57: CUB 128x128 results for Conditional StyleGAN.

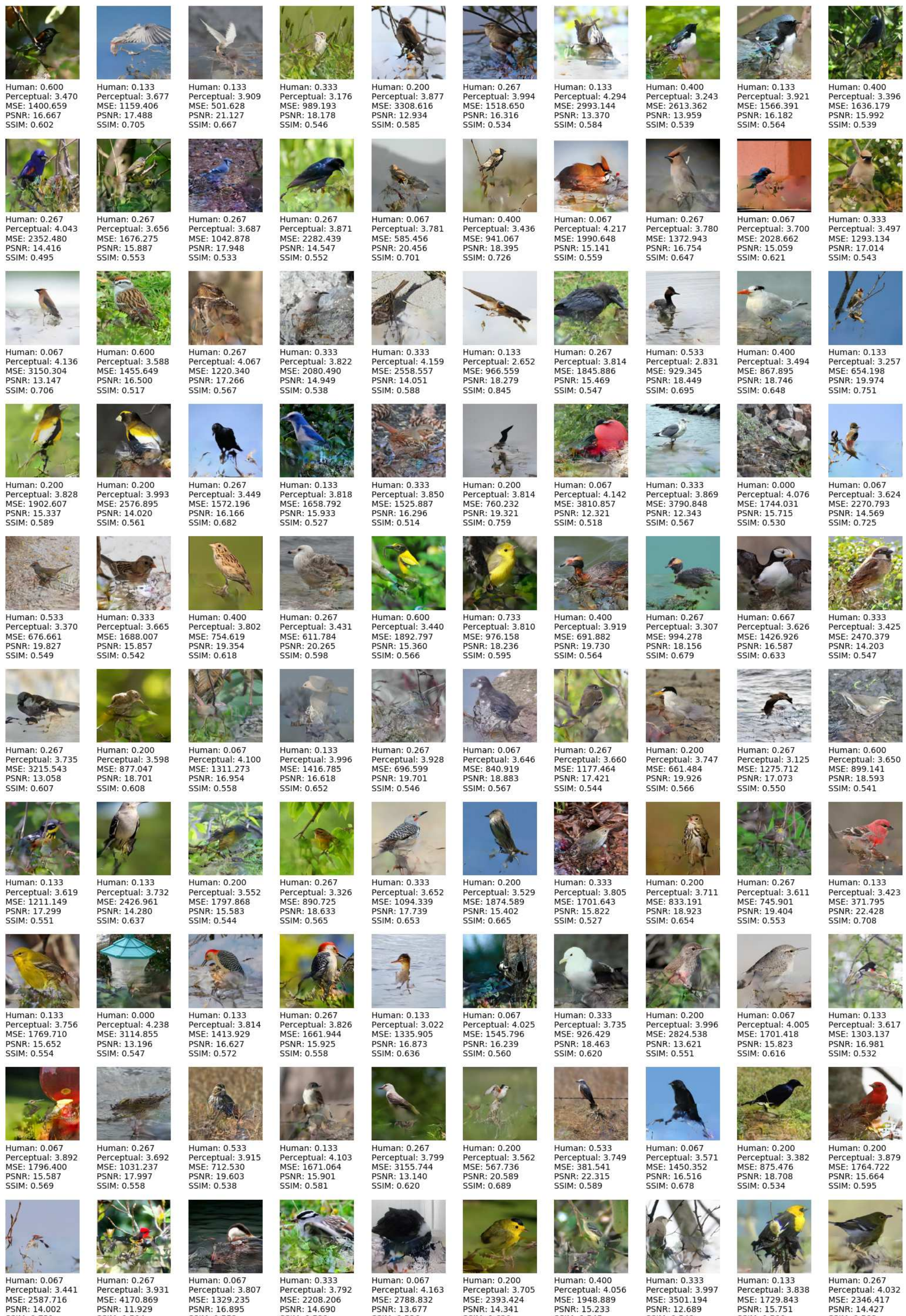

Figure 58: CUB 128x128 results for Conditional ProGAN.

A.4 LSUN-Bedroom

A.4.1 32x32

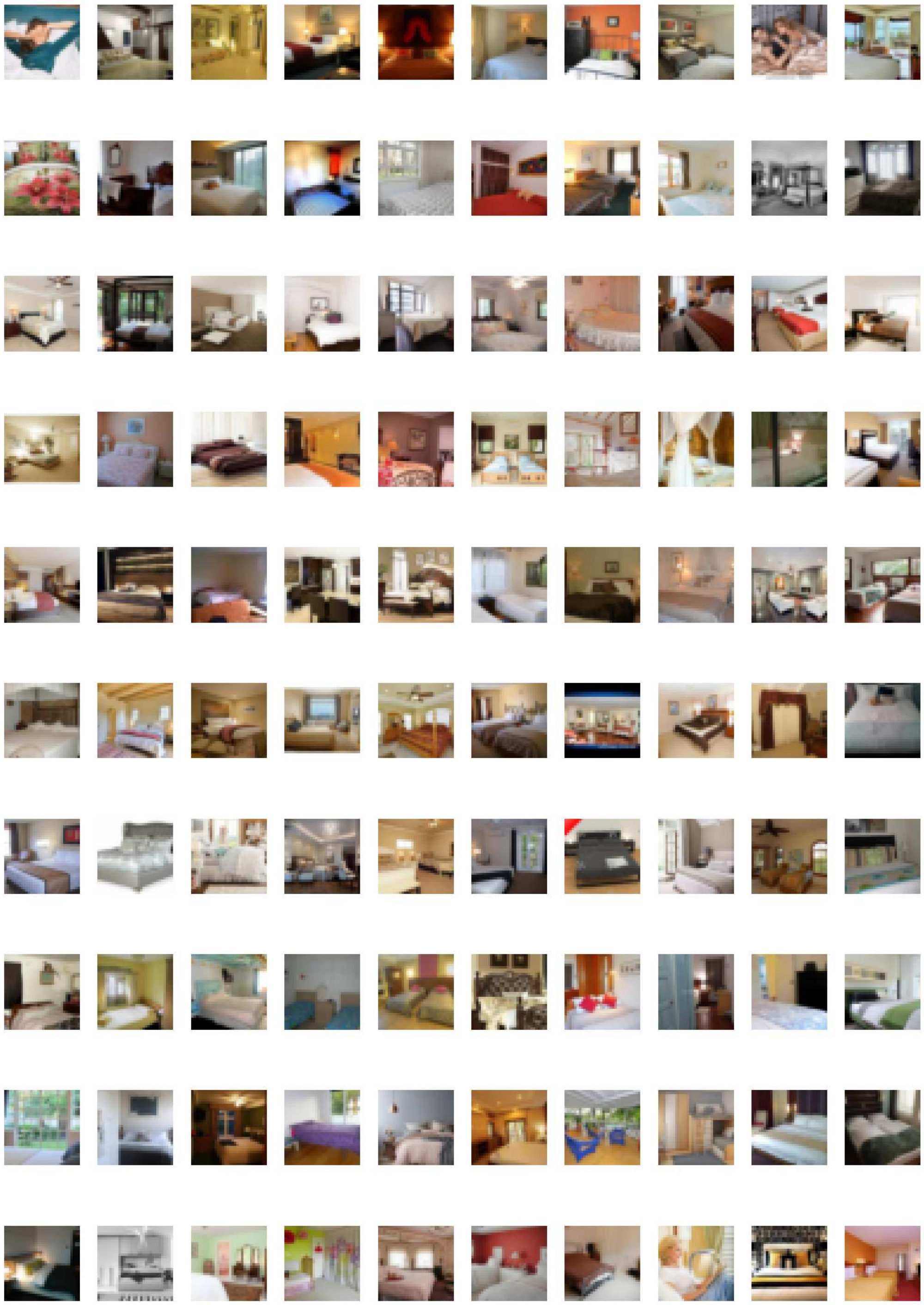

Figure 59: Ground truth for LSUN-Bedroom at 32x32 resolution.

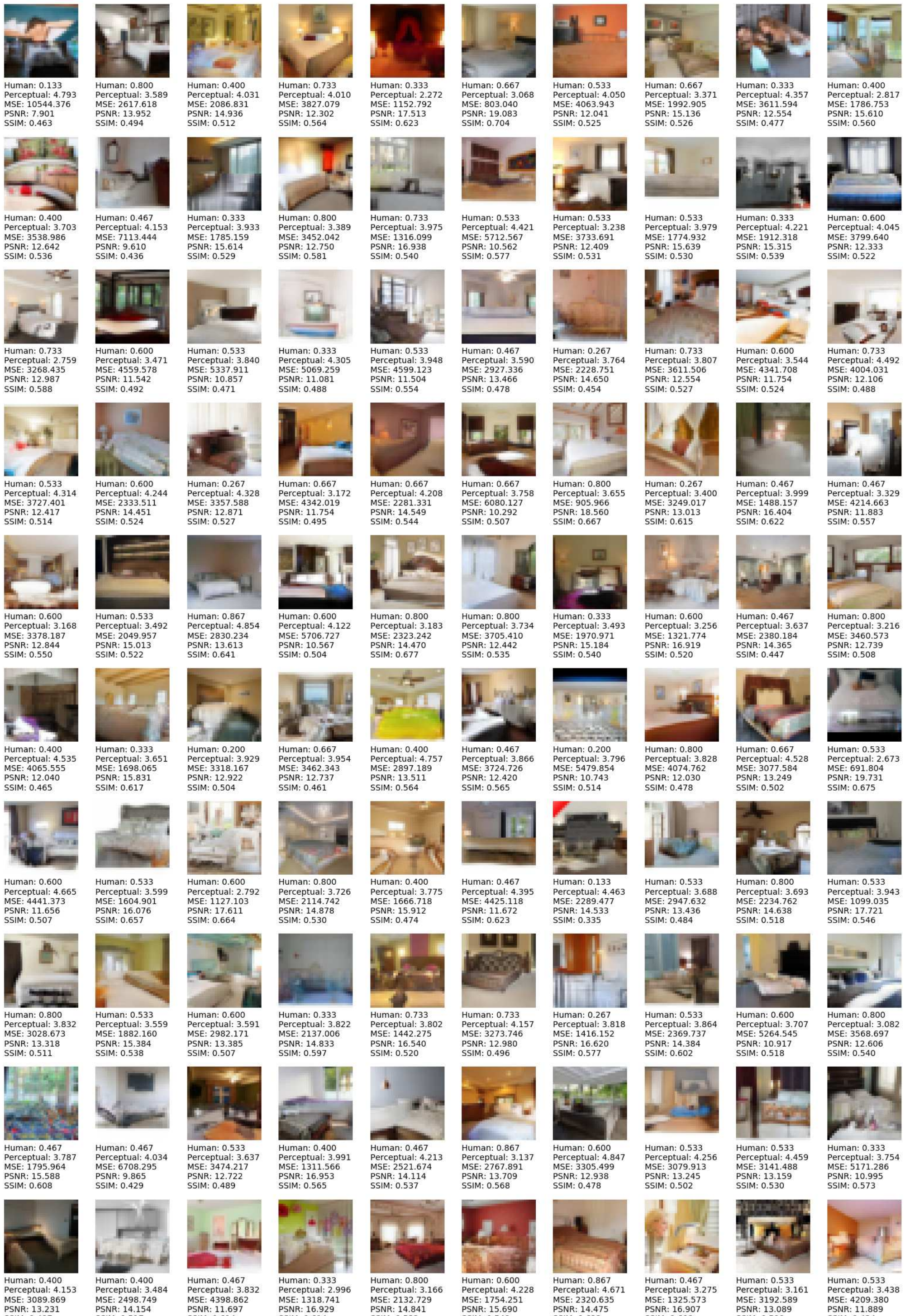

Figure 60: LSUN-Bedroom 32x32 results for ProGAN.

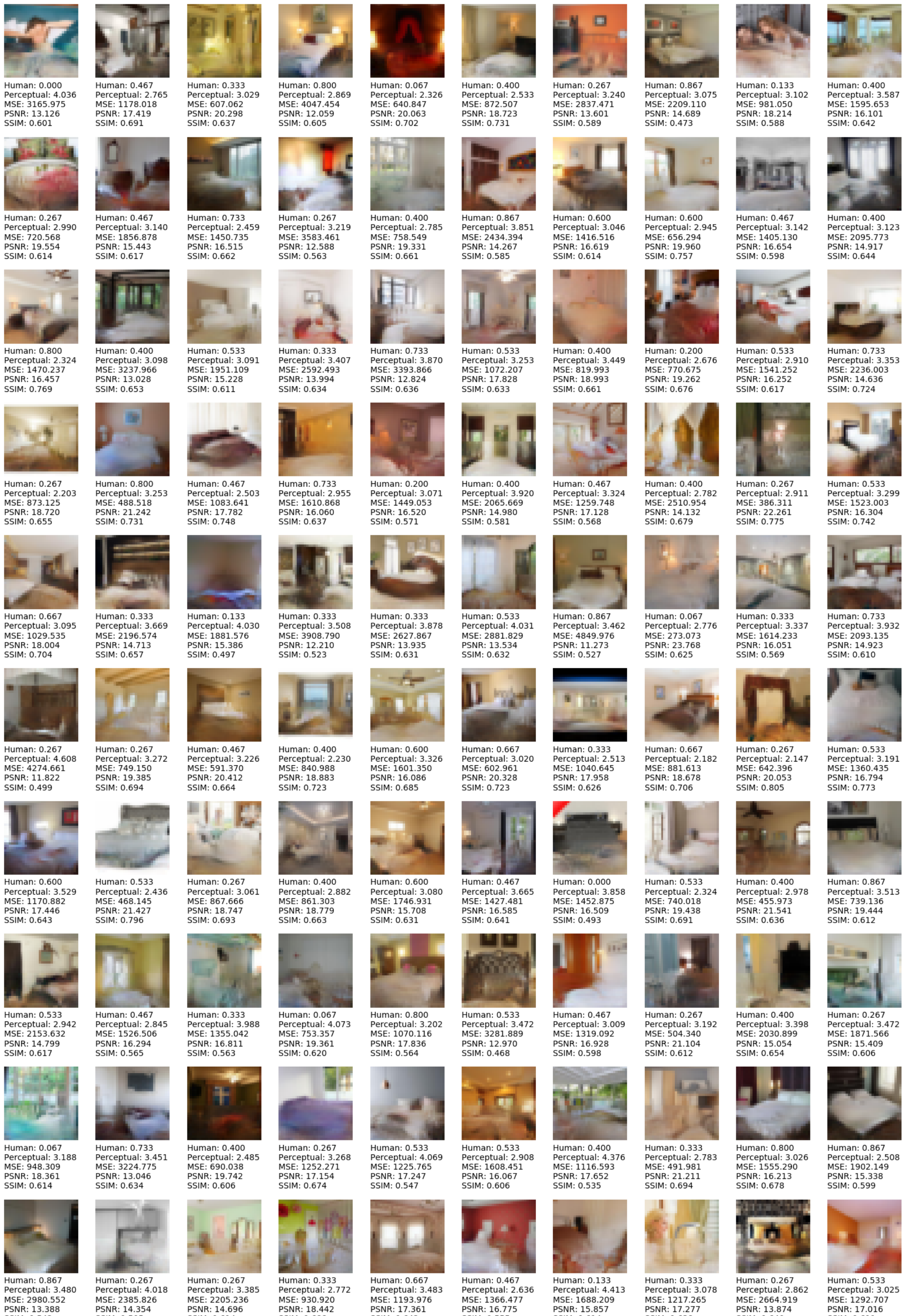

Figure 61: LSUN-Bedroom 32x32 results for Conditional StyleGAN.

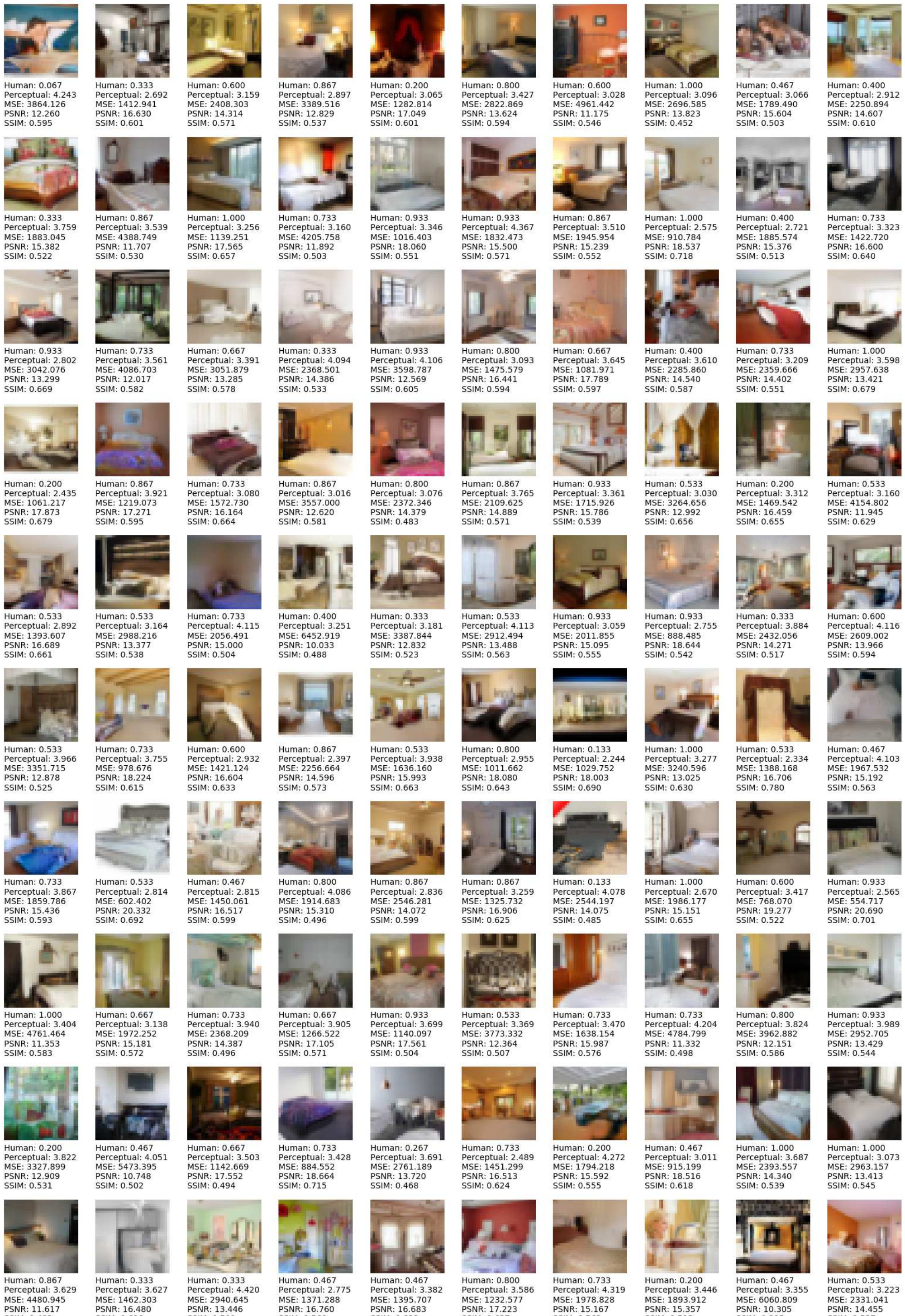

Figure 62: LSUN-Bedroom 32x32 results for Conditional ProGAN.

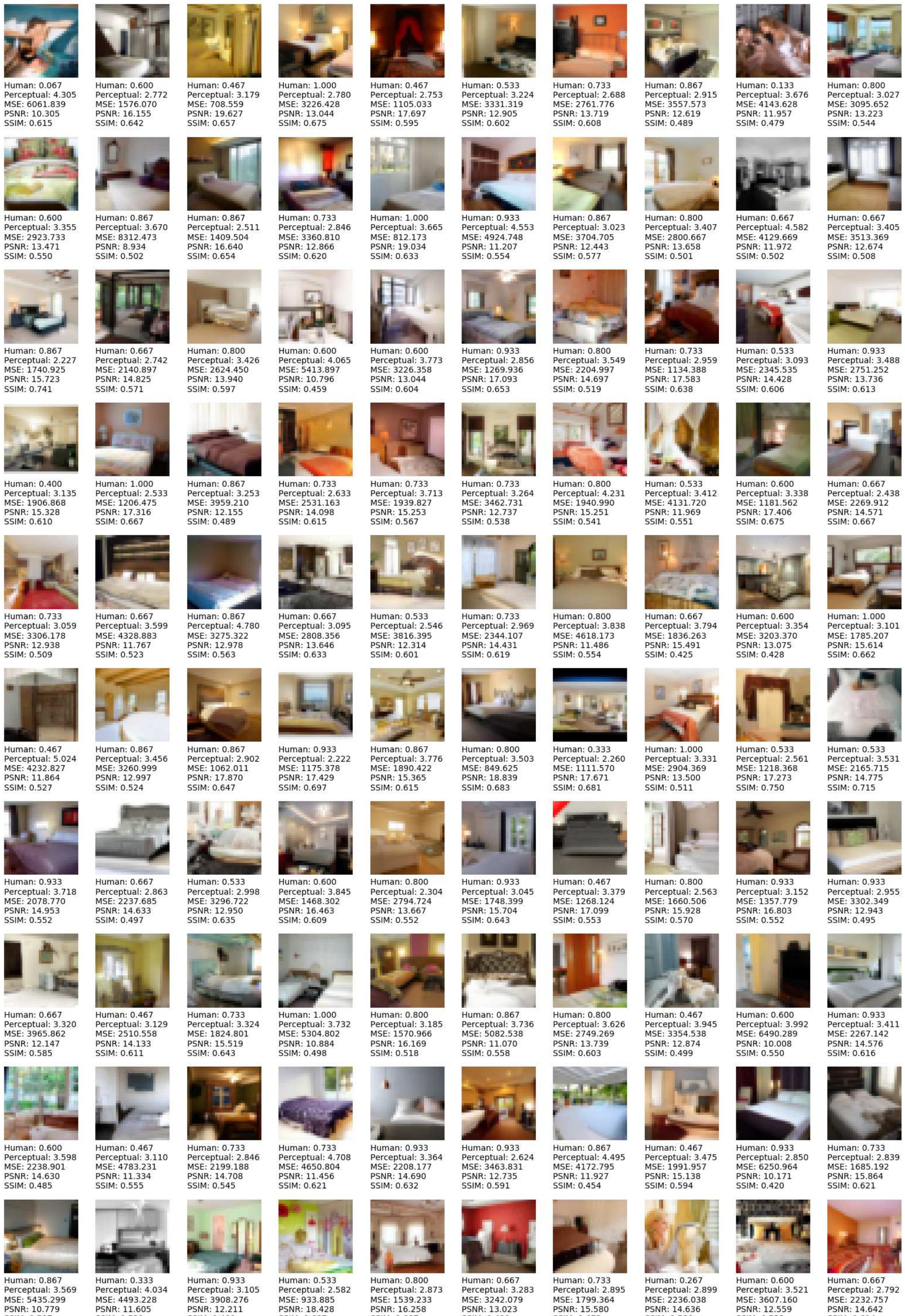

Figure 63: LSUN-Bedroom 32x32 results for PixelCNN++.

A.4.2 64x64

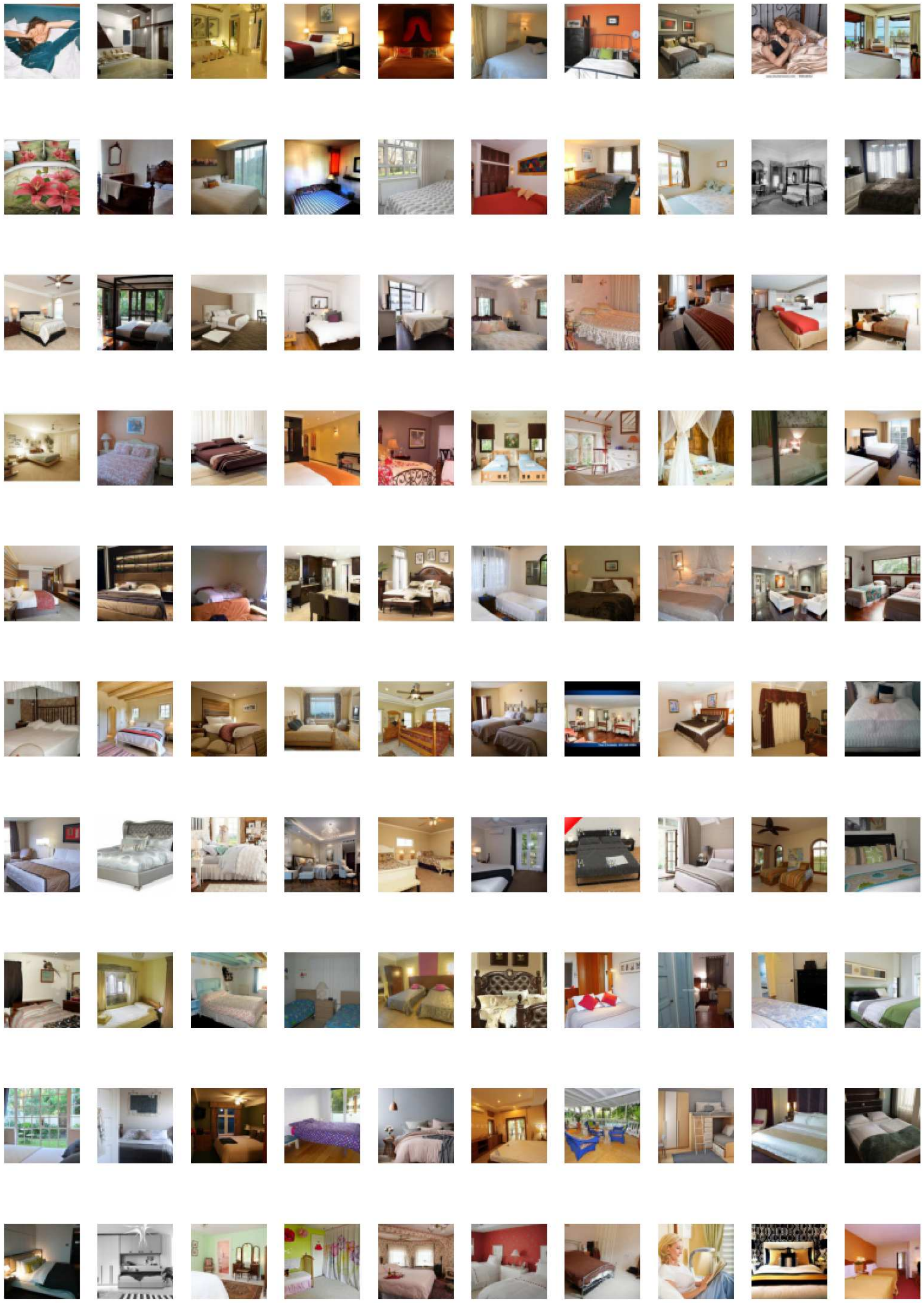

Figure 64: Ground truth for LSUN-Bedroom at 64x64 resolution.

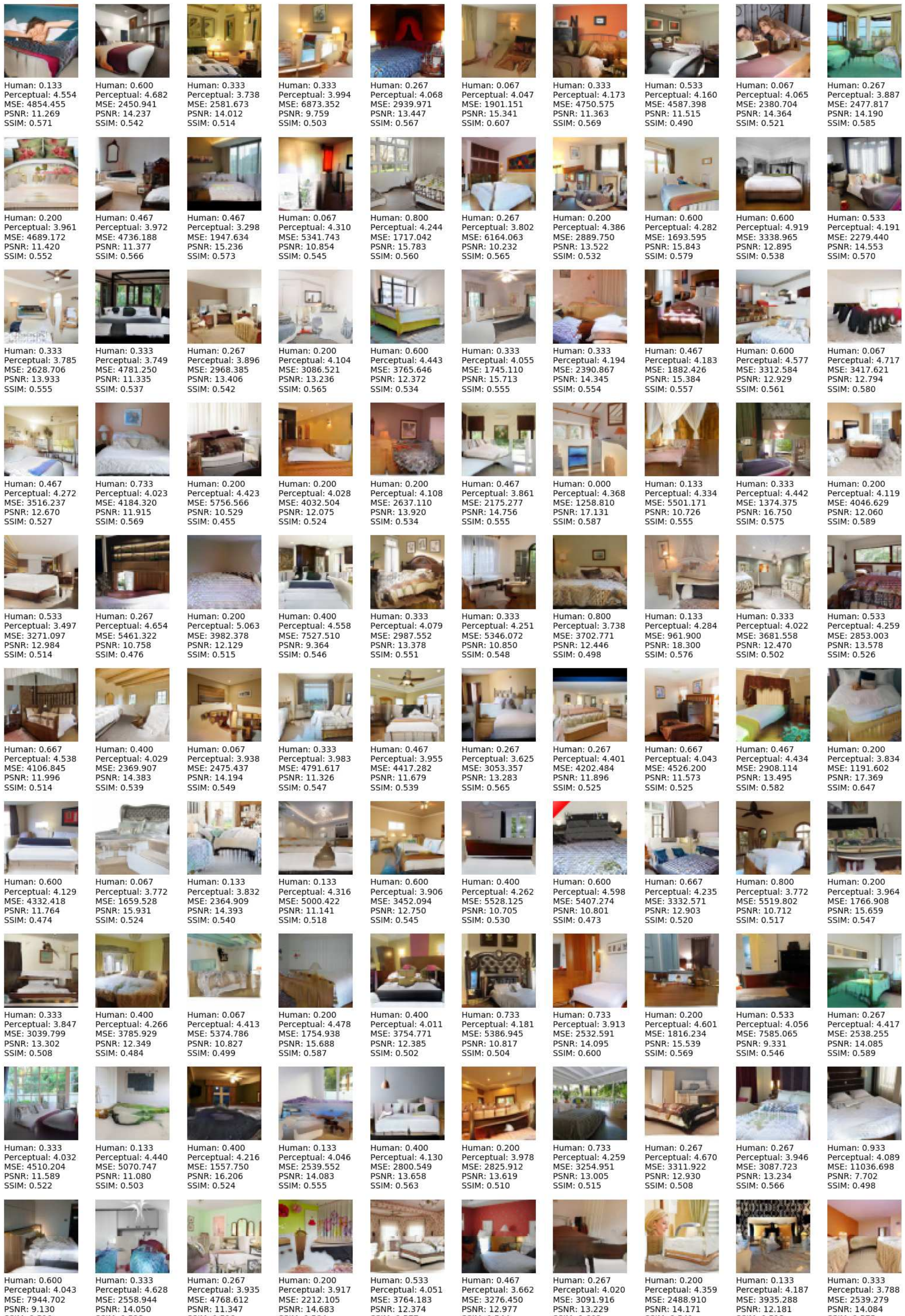

Figure 65: LSUN-Bedroom 64x64 results for ProGAN.

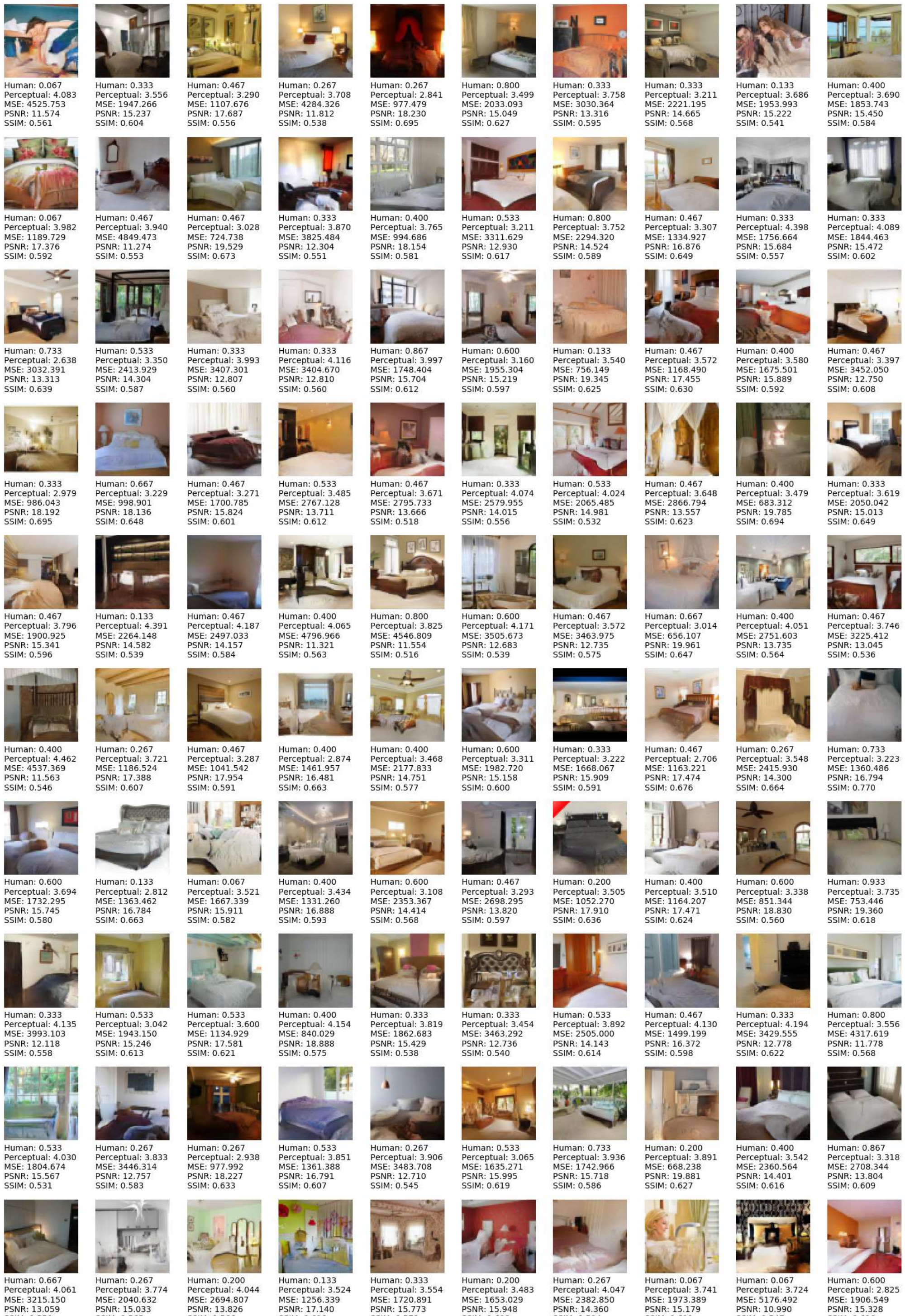

Figure 66: LSUN-Bedroom 64x64 results for Conditional StyleGAN.

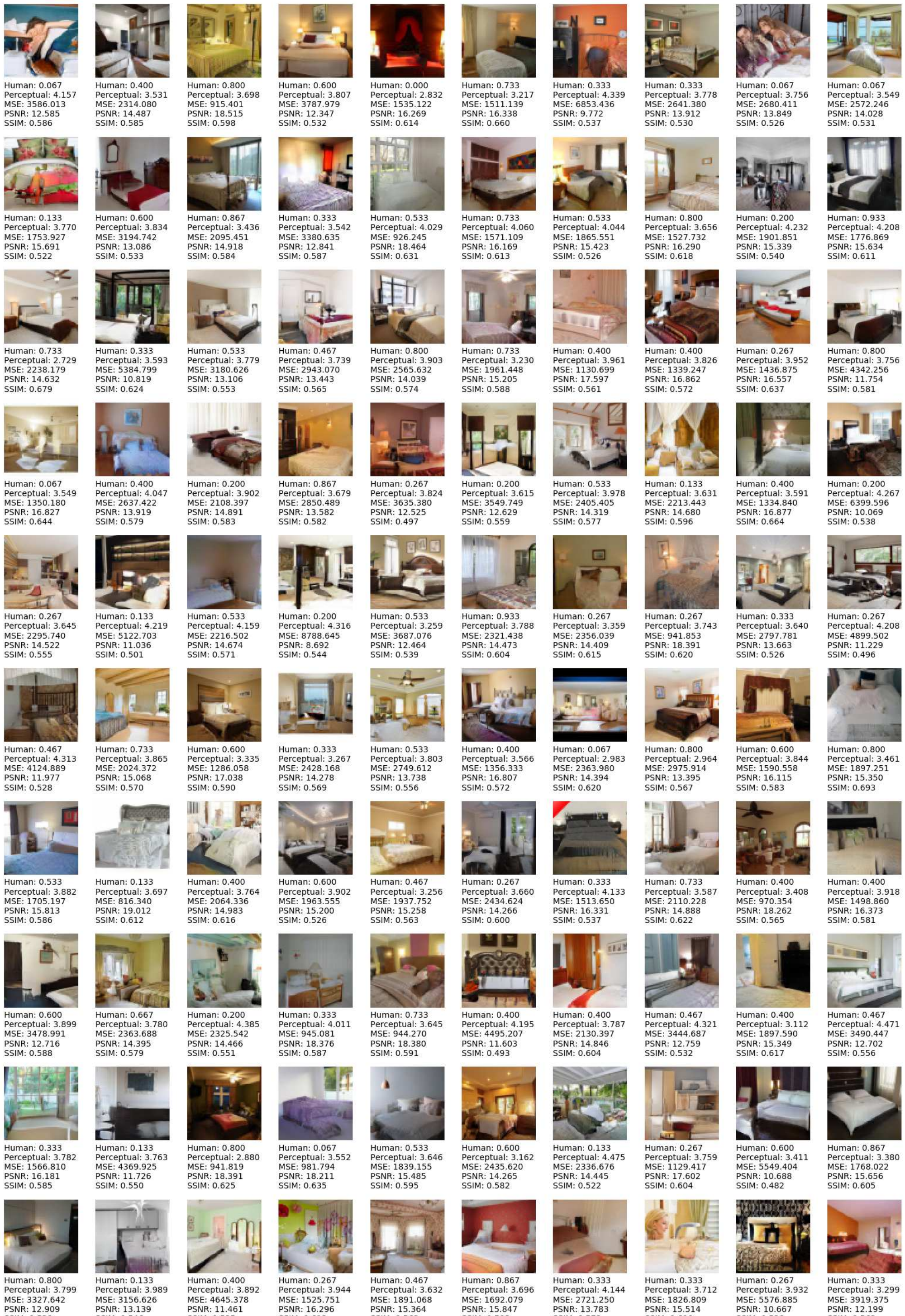

Figure 67: LSUN-Bedroom 64x64 results for Conditional ProGAN.

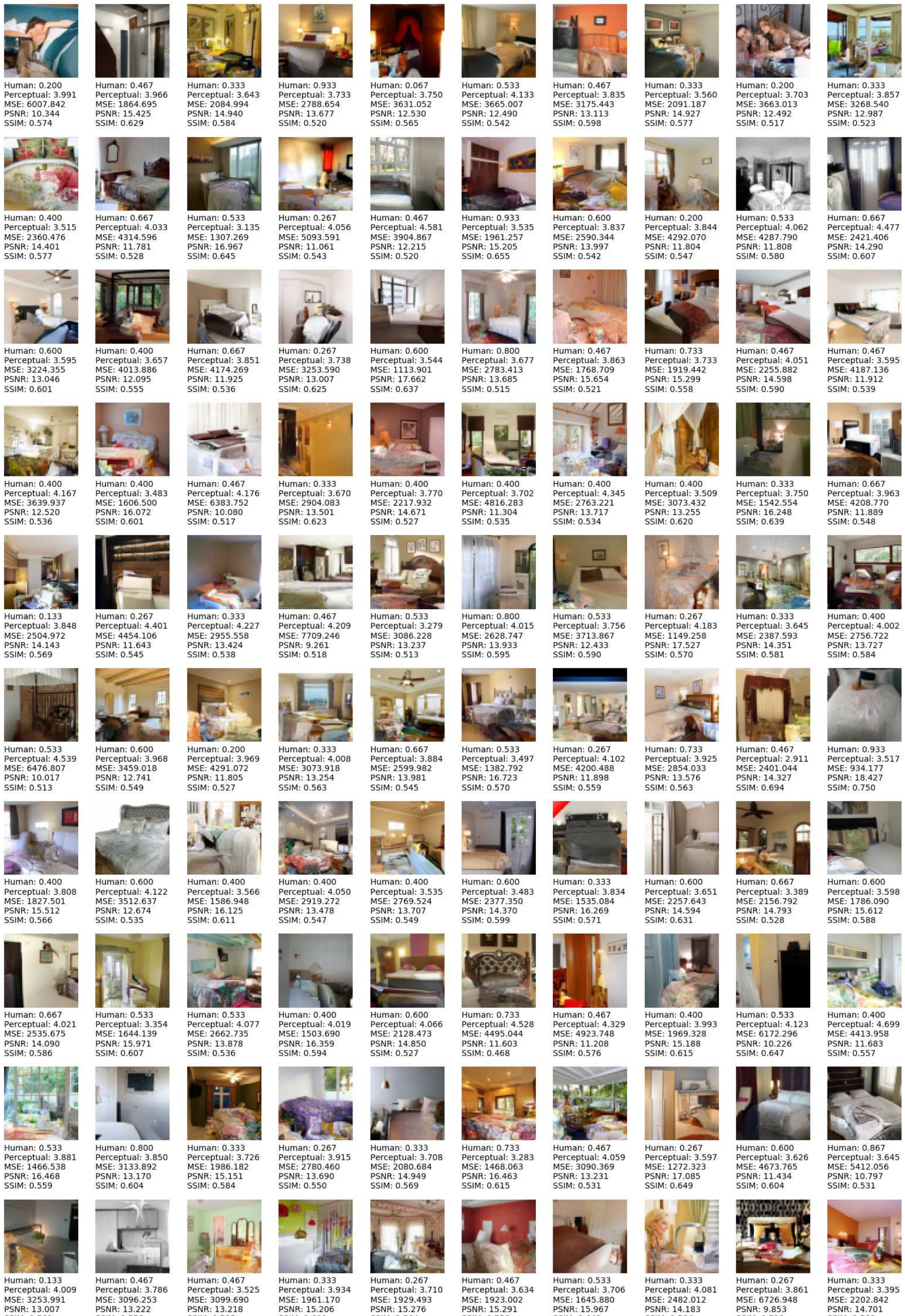

Figure 68: LSUN-Bedroom 64x64 results for PixelCNN++.

A.4.3 128x128

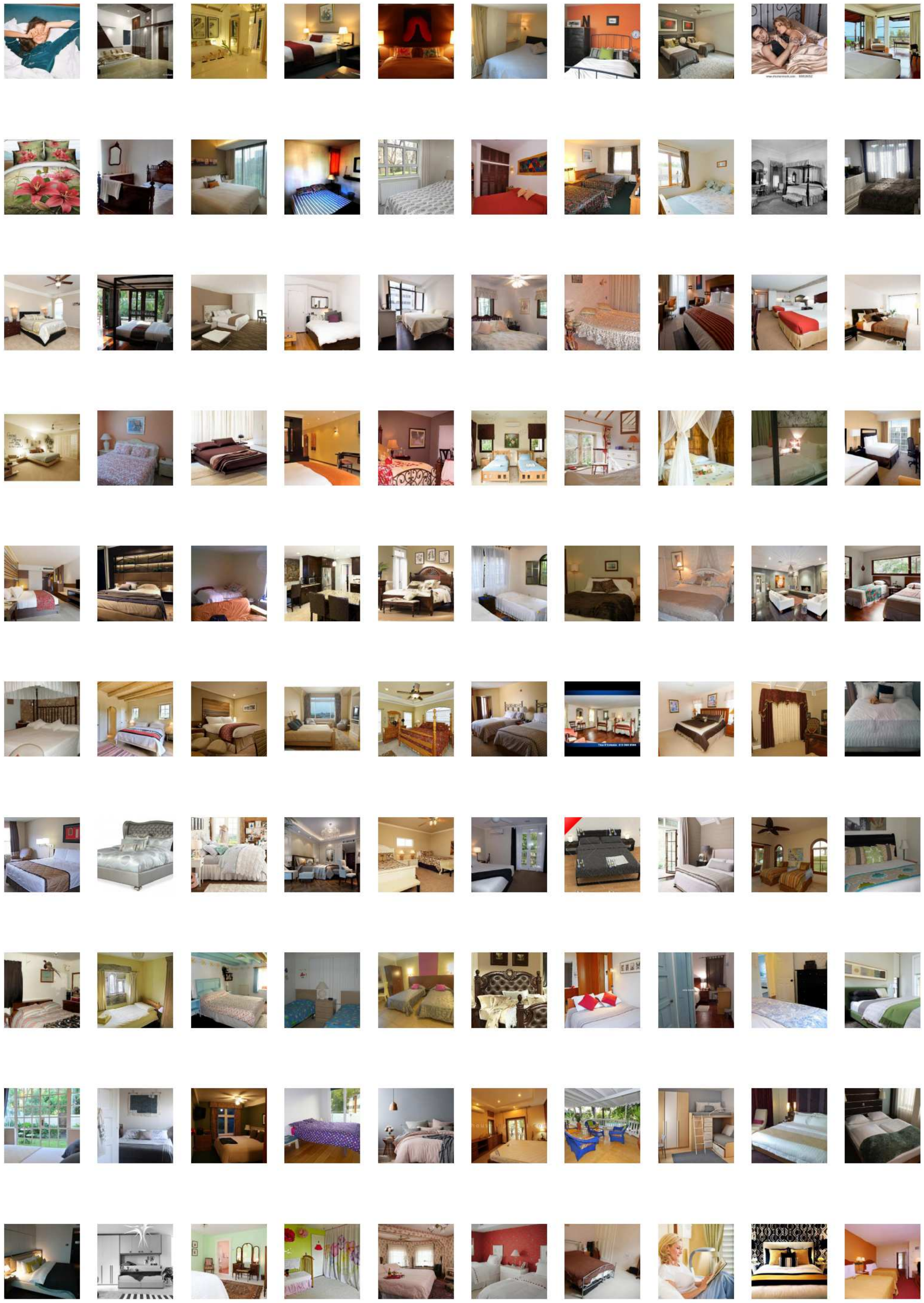

Figure 69: Ground truth for LSUN-Bedroom at 128x128 resolution.

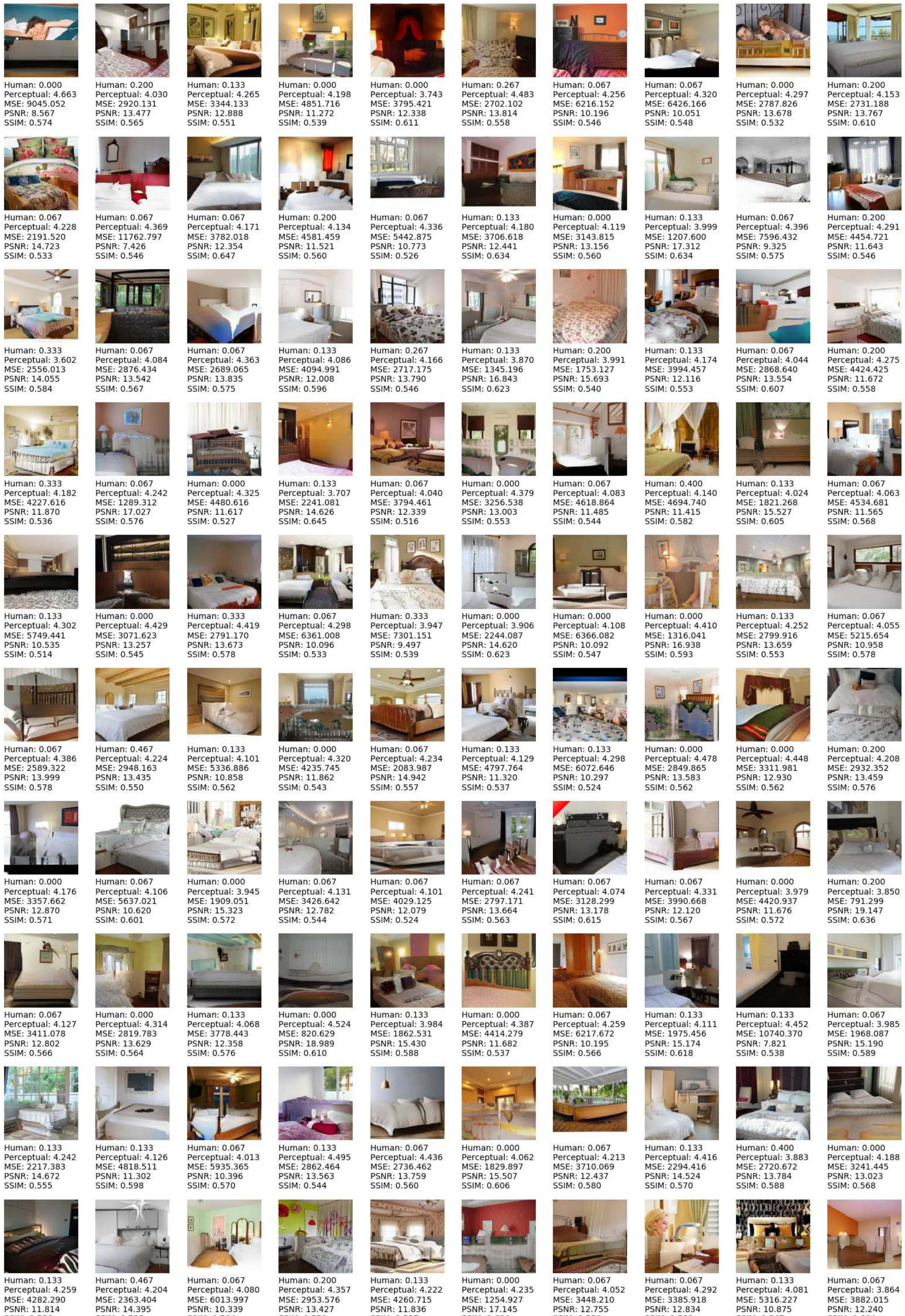

Figure 70: LSUN-Bedroom 128x128 results for ProGAN.

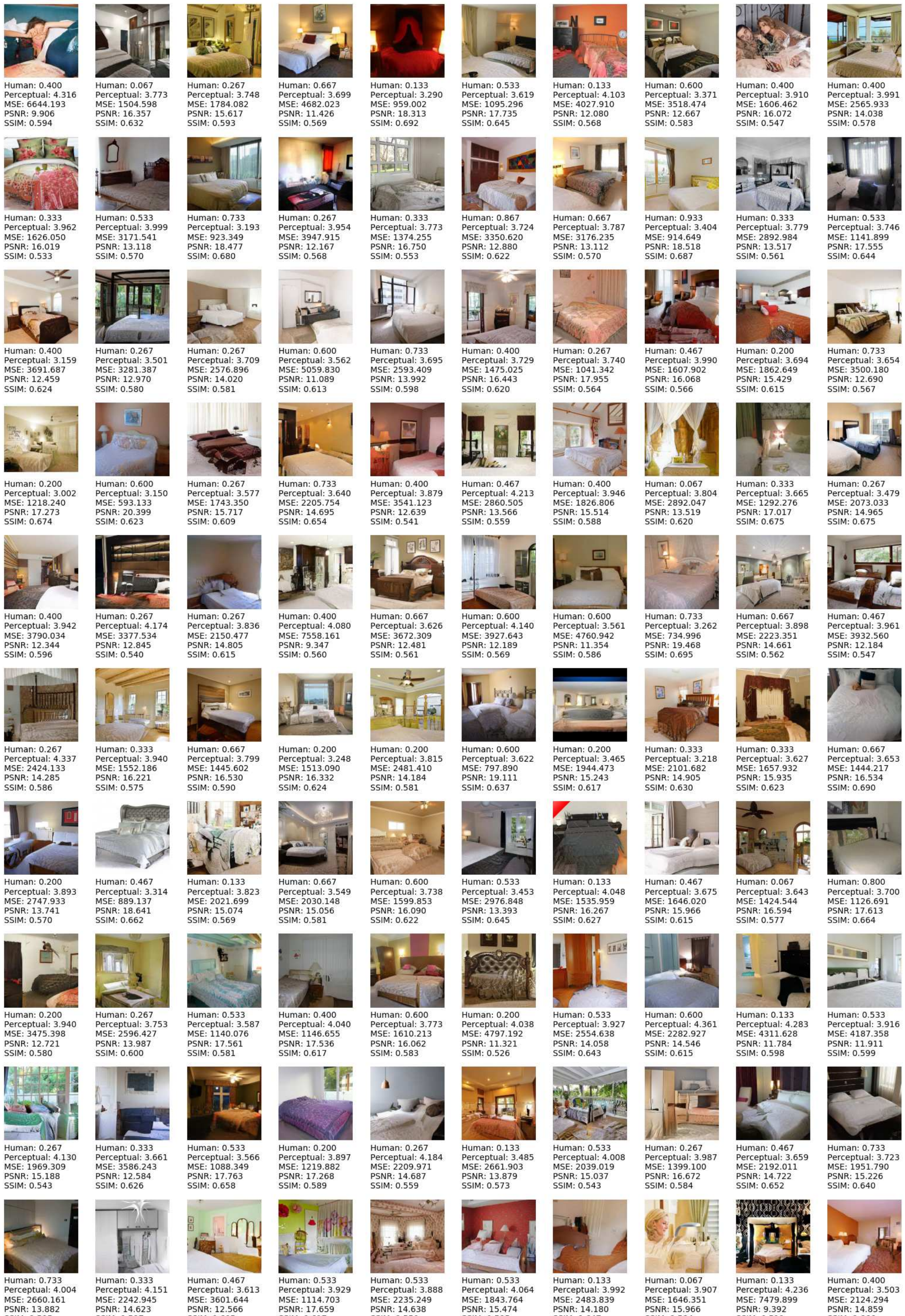

Figure 71: LSUN-Bedroom 128x128 results for Conditional StyleGAN.

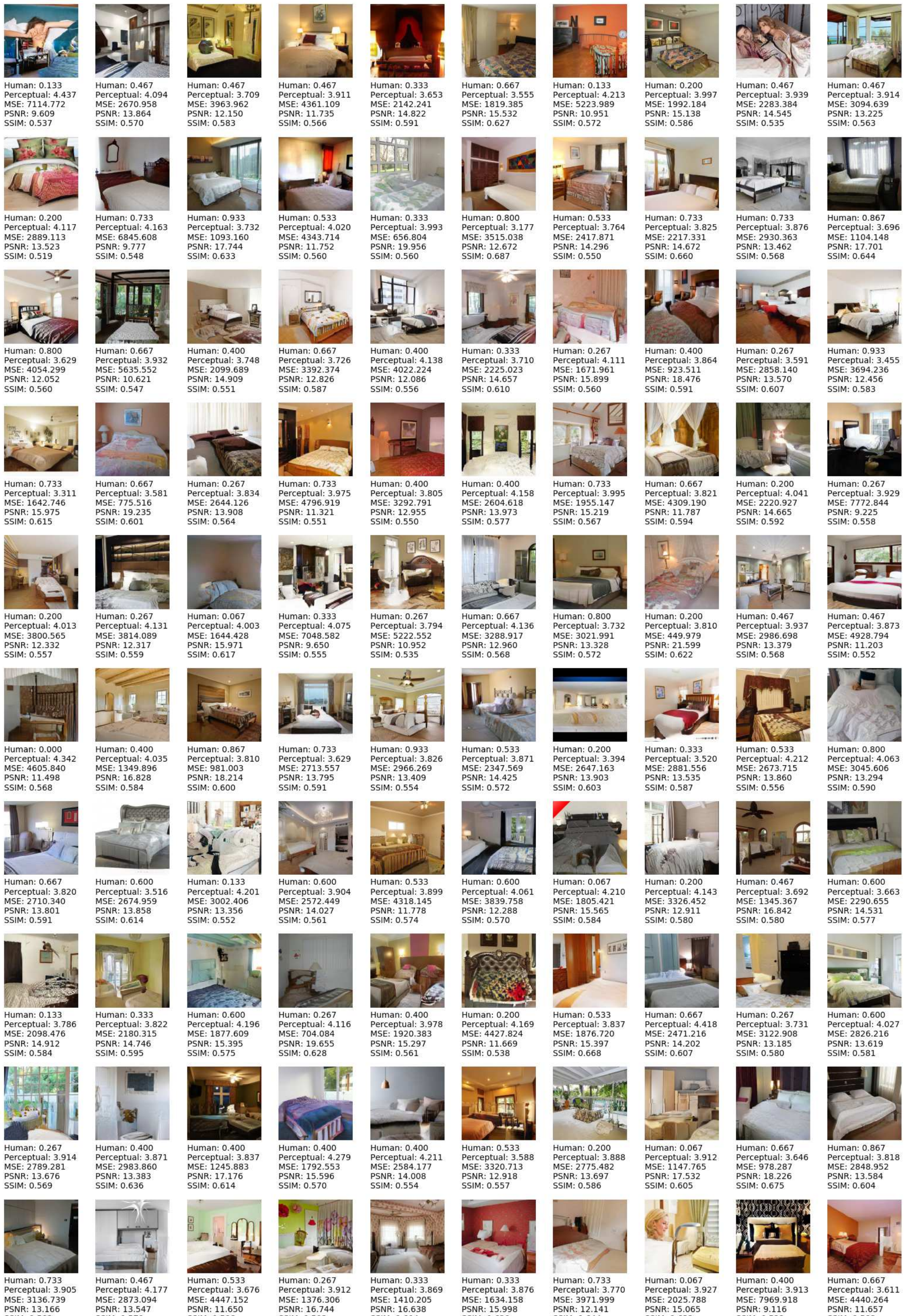

Figure 72: LSUN-Bedroom 128x128 results for Conditional ProGAN.

A.5 LSUN-CAT

A.5.1 32X32

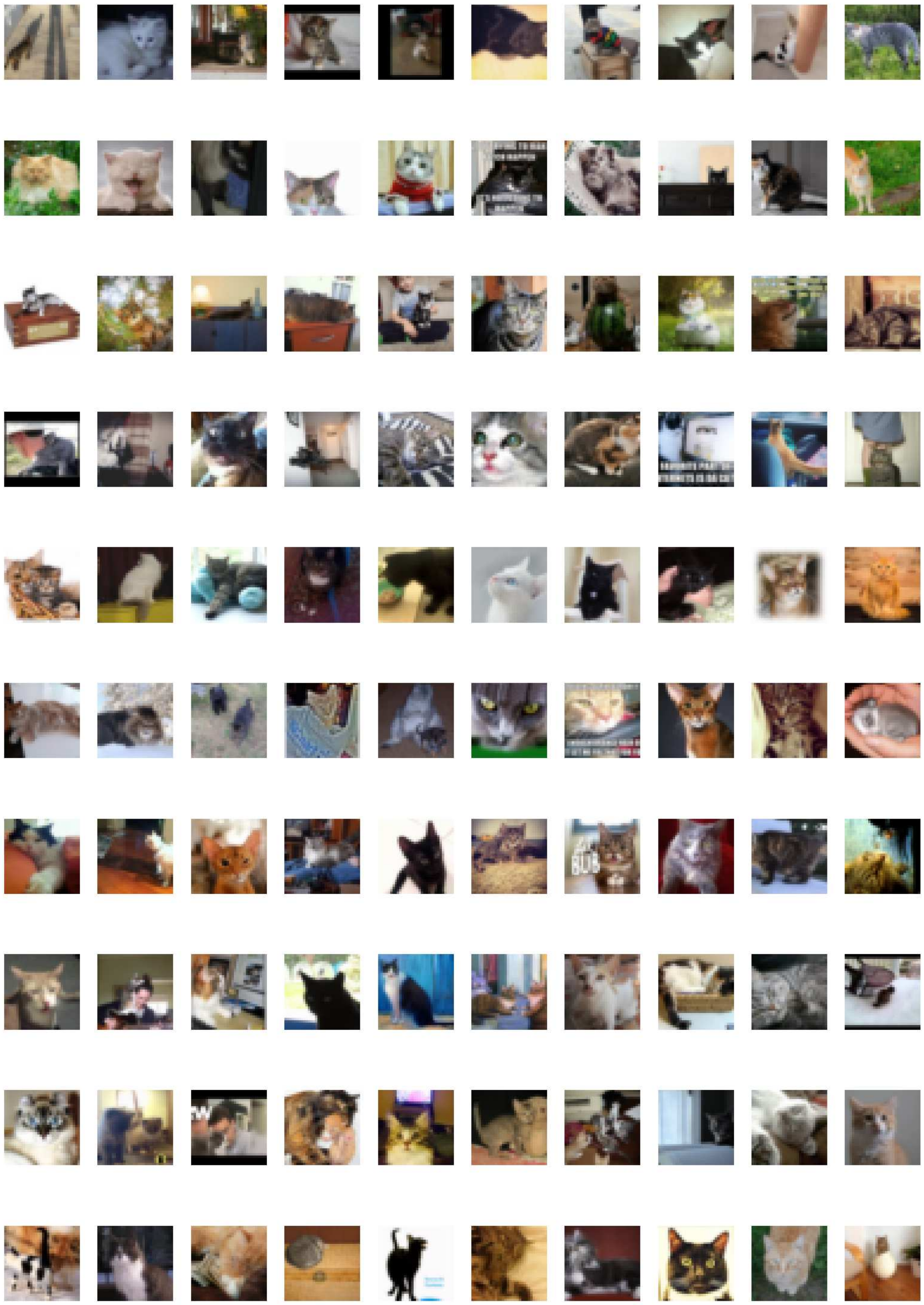

Figure 73: Ground truth for LSUN-Cat at 32x32 resolution.

Figure 74: LSUN-Cat 32x32 results for ProGAN.

Figure 75: LSUN-Cat 32x32 results for Conditional StyleGAN.

Figure 76: LSUN-Cat 32x32 results for Conditional ProGAN.

Figure 77: LSUN-Cat 32x32 results for PixelCNN++.

A.5.2 64x64

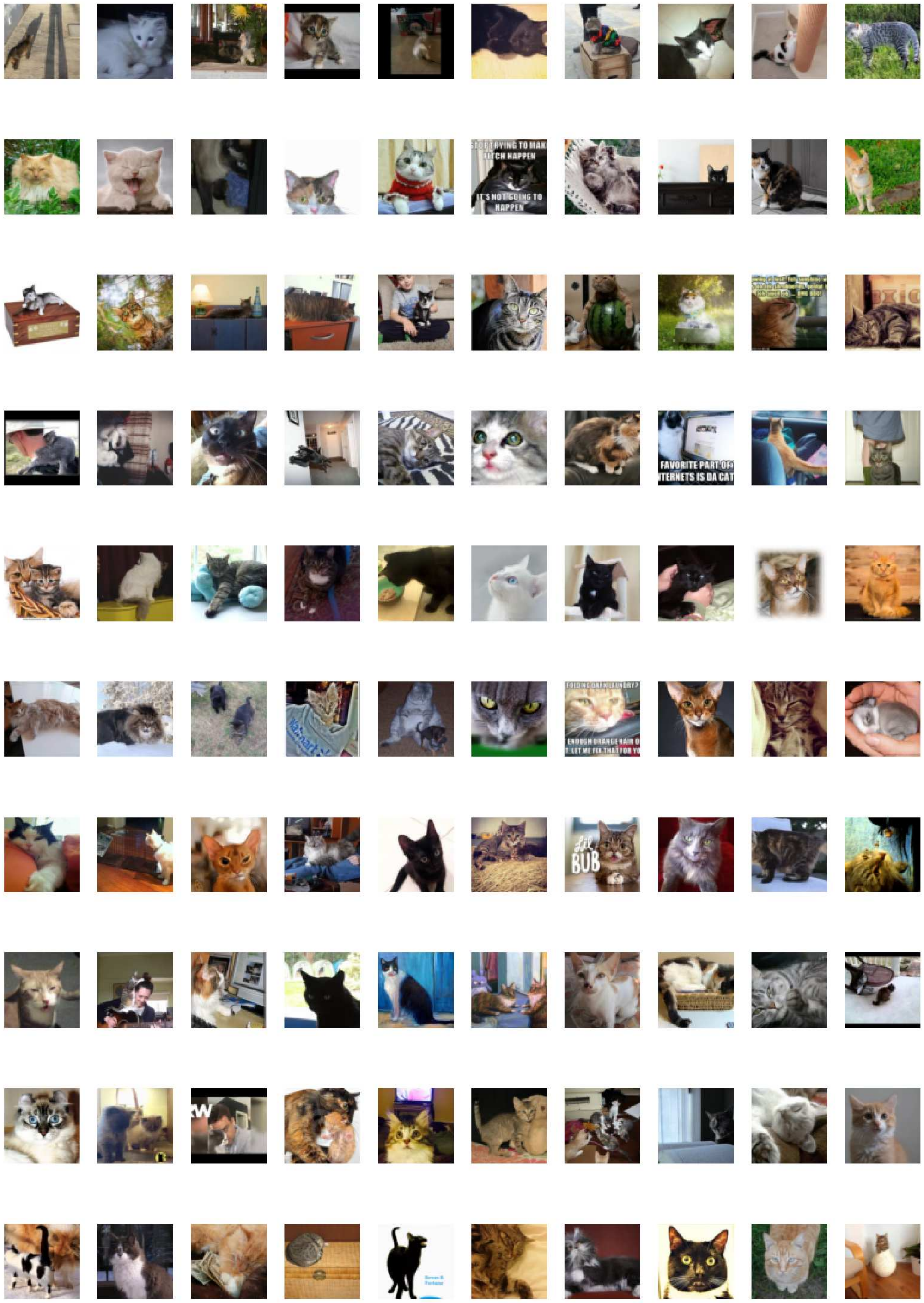

Figure 78: Ground truth for LSUN-Cat at 64x64 resolution.

Figure 79: LSUN-Cat 64x64 results for ProGAN.

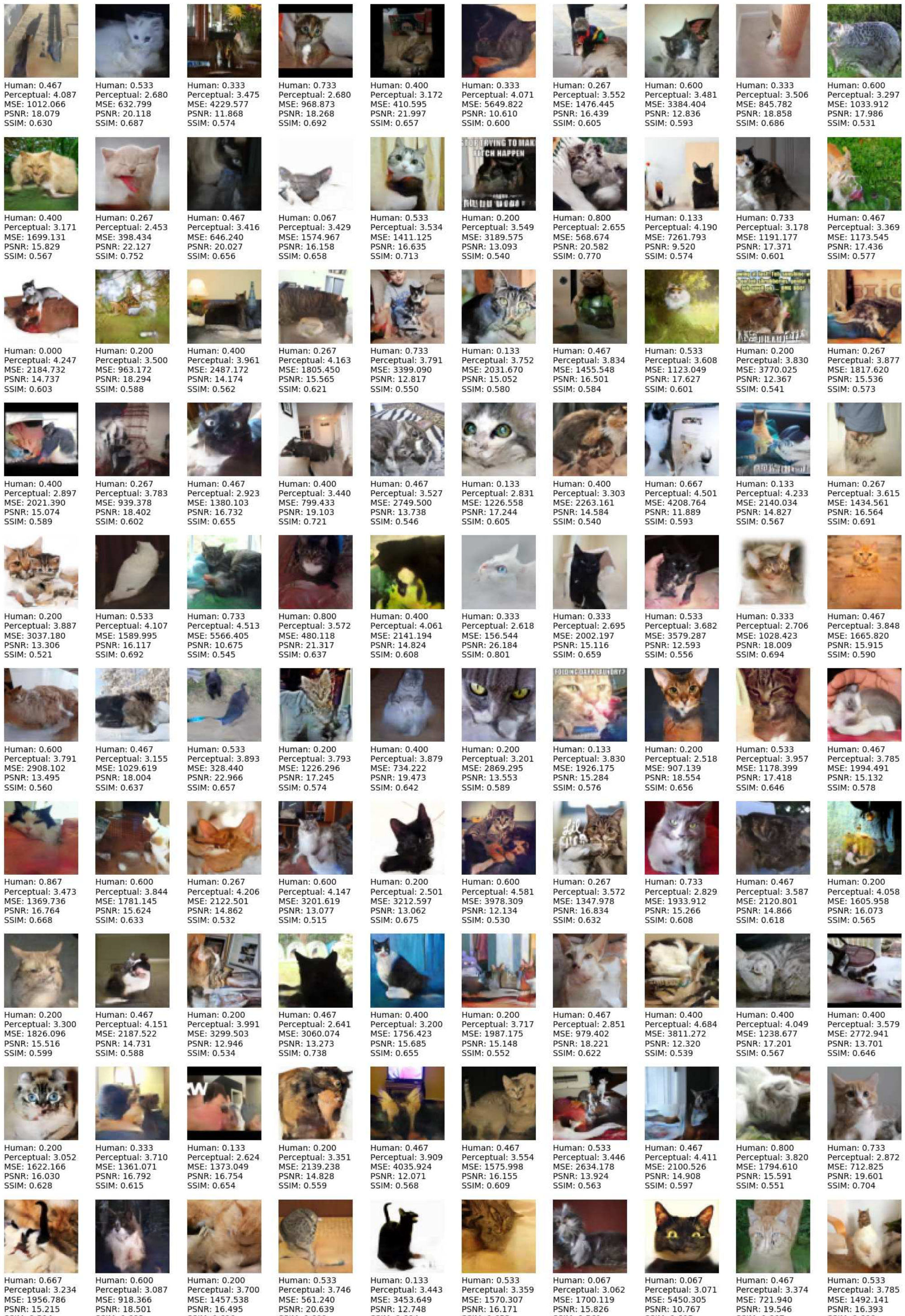

Figure 80: LSUN-Cat 64x64 results for Conditional StyleGAN.

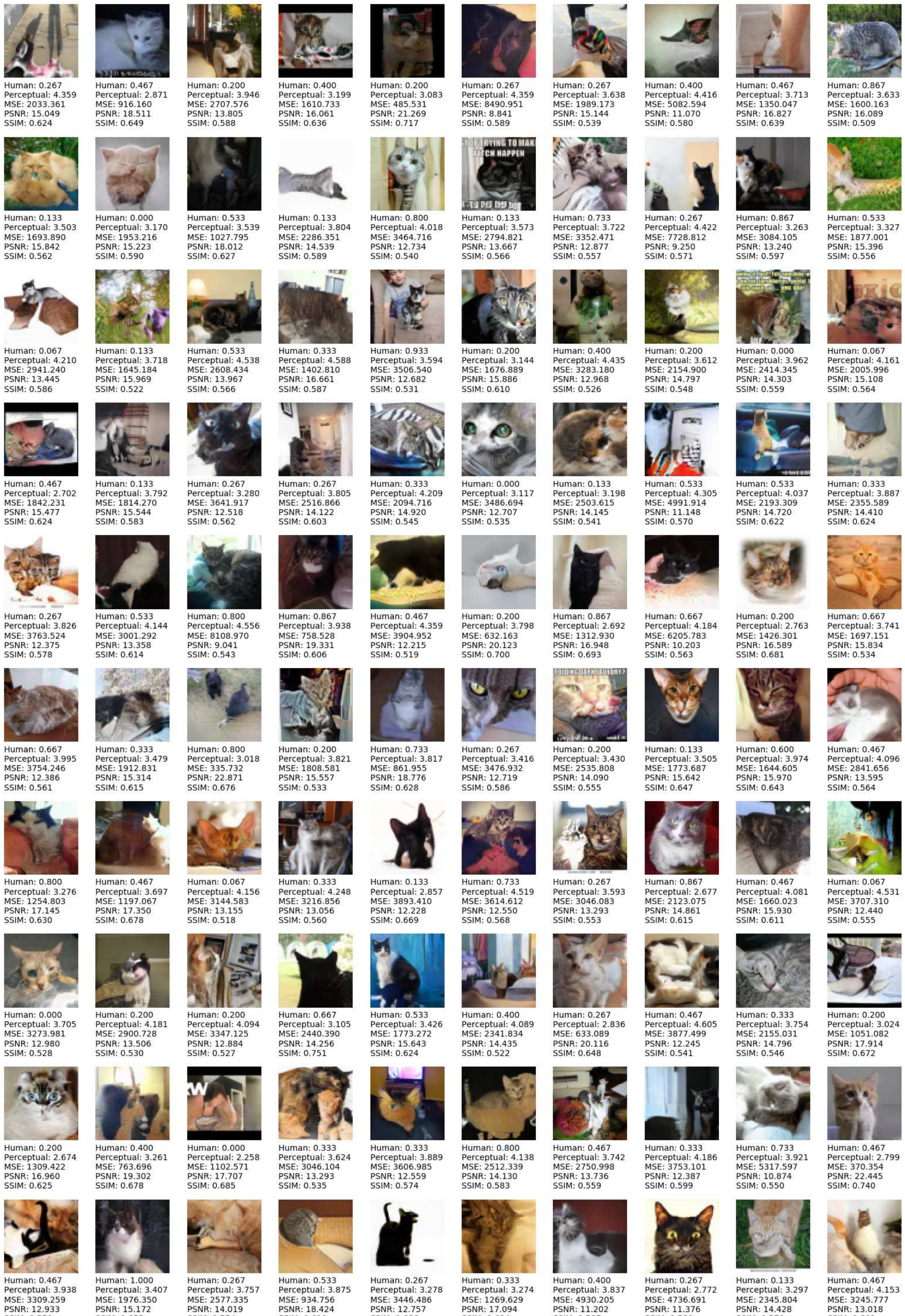

Figure 81: LSUN-Cat 64x64 results for Conditional ProGAN.

Figure 82: LSUN-Cat 64x64 results for PixelCNN++.

A.5.3 128x128

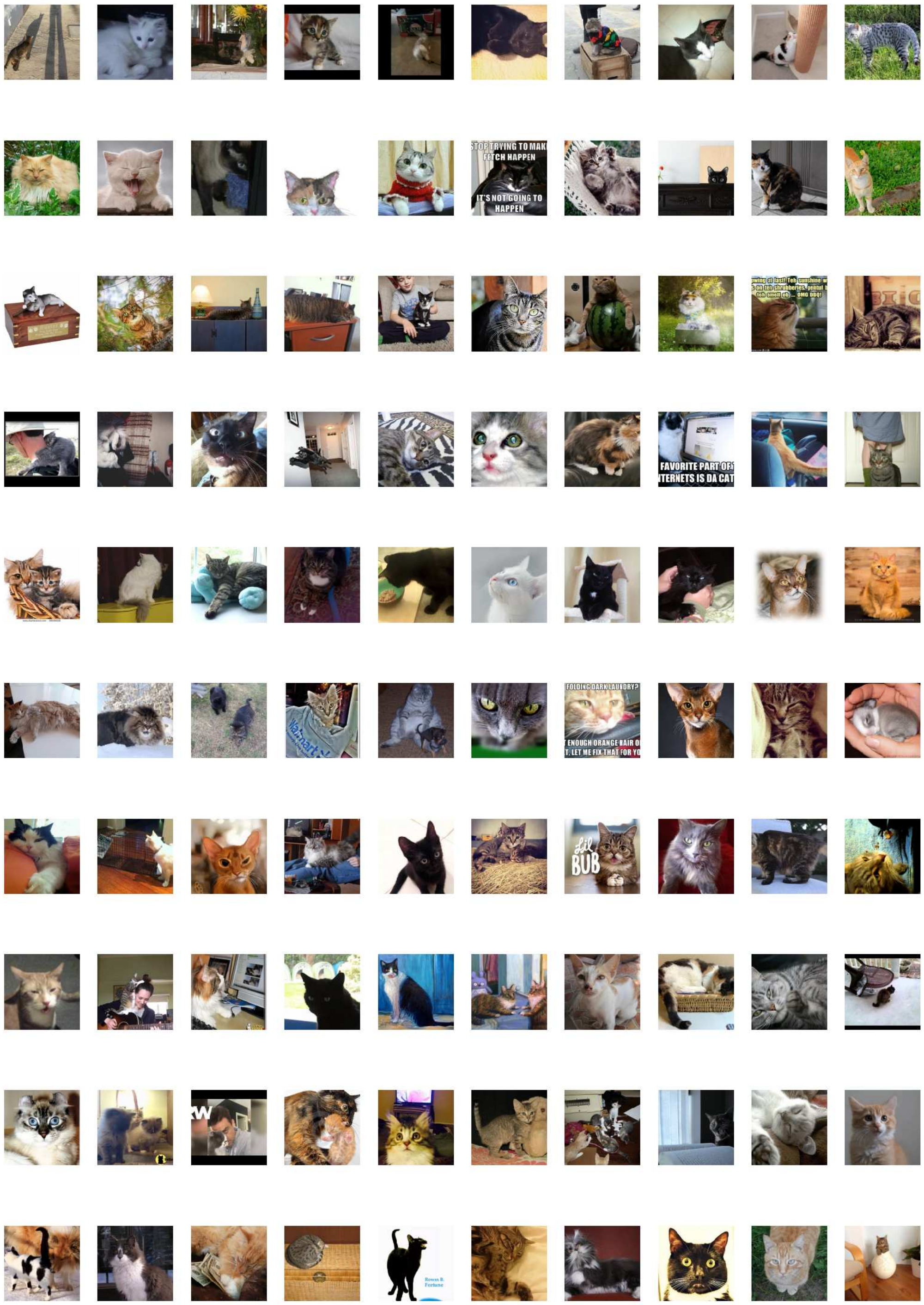

Figure 83: Ground truth for LSUN-Cat at 128x128 resolution.

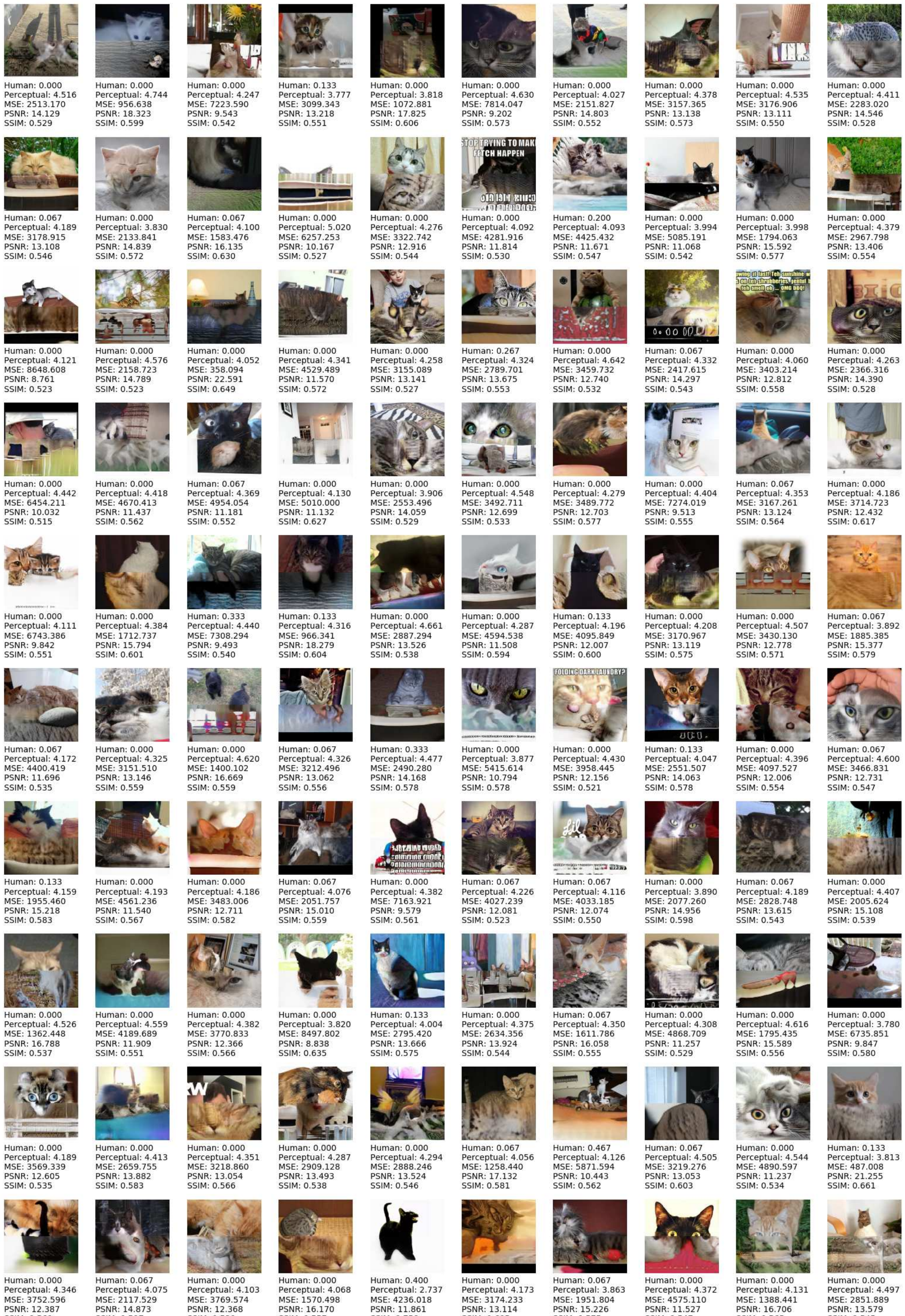

Figure 84: LSUN-Cat 128x128 results for ProGAN.

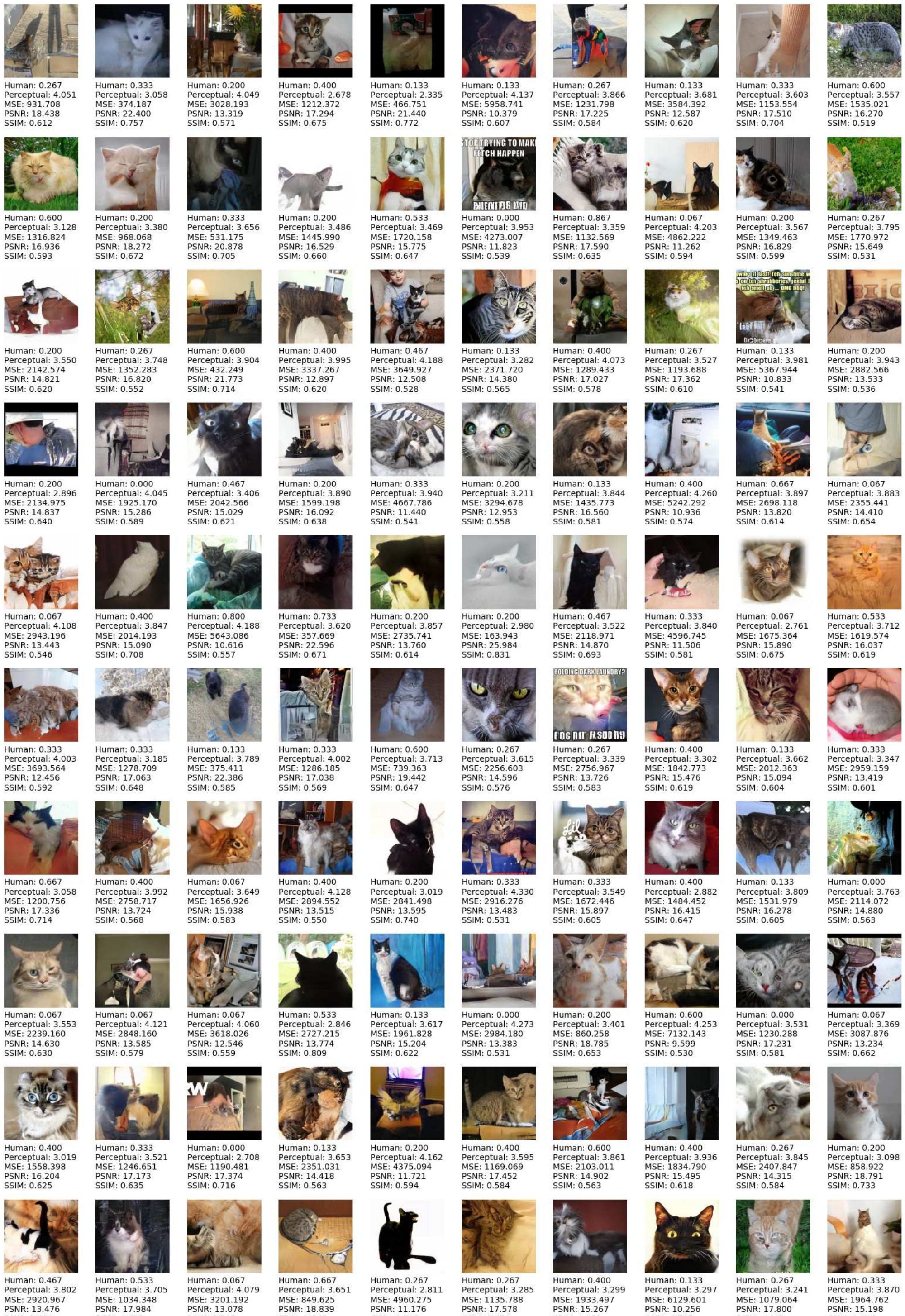

Figure 85: LSUN-Cat 128x128 results for Conditional StyleGAN.

Figure 86: LSUN-Cat 128x128 results for Conditional ProGAN.