Pixel Recursive Super Resolution

Ryan Dahl * Mohammad Norouzi Jonathon Shlens

Google Brain

{rld,mnorouzi,shlens}@google.com

Abstract

We present a pixel recursive super resolution model that synthesizes realistic details into images while enhancing their resolution. A low resolution image may correspond to multiple plausible high resolution images, thus modeling the super resolution process with a pixel independent conditional model often results in averaging different details–hence blurry edges. By contrast, our model is able to represent a multimodal conditional distribution by properly modeling the statistical dependencies among the high resolution image pixels, conditioned on a low resolution input. We employ a PixelCNN architecture to define a strong prior over natural images and jointly optimize this prior with a deep conditioning convolutional network. Human evaluations indicate that samples from our proposed model look more photo realistic than a strong L2 regression baseline.

1. Introduction

The problem of *super resolution* entails artificially enlarging a low resolution photograph to recover a plausible high resolution version of it. When the zoom factor is large, the input image does not contain all of the information necessary to accurately construct a high resolution image. Thus, the problem is underspecified and many plausible high resolution images exist that match the low resolution input image. This problem is significant for improving the state-of-the-art in super resolution, and more generally for building better conditional generative models of images.

A super resolution model must account for the complex variations of objects, viewpoints, illumination, and occlusions, especially as the zoom factor increases. When some details do not exist in the source image, the challenge lies not only in 'deblurring' an image, but also in generating new image details that appear plausible to a human observer. Generating realistic high resolution images is not possible unless the model draws sharp edges and makes hard decisions about the type of textures, shapes, and patterns present at different parts of an image.

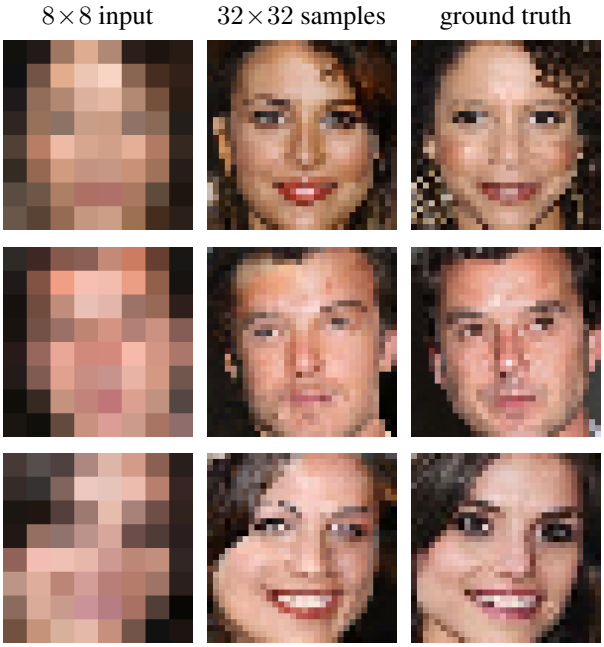


Figure 1: Illustration of our probabilistic pixel recursive super resolution model trained end-to-end on a dataset of celebrity faces. The left column shows 8×8 low resolution inputs from the test set. The middle and last columns show 32×32 images as predicted by our model *vs.* the ground truth. Our model incorporates strong face priors to synthesize realistic hair and skin details.

Imagine a low resolution image of a face, *e.g.,* the 8×8 images depicted in the left column of Figure 1–the details

*Work done as a member of the Google Brain Residency program (g.co/brainresidency).

of the hair and the skin are missing. Such details cannot be faithfully recovered using simple interpolation techniques such as linear or bicubic. However, by incorporating the prior knowledge of the faces and their typical variations, an artist is able to paint believable details. In this paper, we show how a fully *probabilistic* model that is trained *end-to-end* can play the role of such an artist by synthesizing 32×32 face images depicted in the middle column of Figure 1. Our super resolution model comprises two components that are trained jointly: *a conditioning* network, and a *prior* network. The conditioning network effectively maps a low resolution image to a distribution over corresponding high resolution images, while the prior models high resolution details to make the outputs look more realistic. Our conditioning network consists of a deep stack of ResNet [10] blocks, while our prior network comprises a PixelCNN [28] architecture.

We find that standard super resolution metrics such as peak signal-to-noise ratio (pSNR) and structural similarity (SSIM) fail to properly measure the quality of predictions for an underspecified super resolution task. These metrics prefer conservative blurry averages over more plausible photo realistic details, as new fine details often do not align exactly with the original details. Our evaluation studies demonstrate that humans easily distinguish real images from super resolution predictions when regression techniques are used, but they have a harder time telling our samples apart from real images.

2. Related work

Super resolution has a long history in computer vision [22]. Methods relying on interpolation [11] are easy to implement and widely used, however these methods suffer from a lack of expressivity since linear models cannot express complex dependencies between the inputs and outputs. In practice, such methods often fail to adequately predict high frequency details leading to blurry high resolution outputs.

Enhancing linear methods with rich image priors such as sparsity [2] or Gaussian mixtures [35] have substantially improved the quality of the methods; likewise, leveraging low-level image statistics such as edge gradients improves predictions [31, 26, 6, 12, 25, 17]. Much work has been done on algorithms that search a database of patches and combine them to create plausible high frequency details in zoomed images [7, 13]. Recent patch-based work has focused on improving basic interpolation methods by building a dictionary of pre-learned filters on images and selecting the appropriate patches by an efficient hashing mechanism [23]. Such dictionary methods have improved the inference speed while being comparable to state-of-the-art.

Another approach for super resolution is to abandon inference speed requirements and focus on constructing the high resolution images at increasingly higher magnification factors. Convolutional neural networks (CNNs) represent an approach to the problem that avoids explicit dictionary construction, but rather implicitly extracts multiple layers of abstractions by learning layers of filter kernels. Dong *et al.* [5] employed a three layer CNN with MSE loss. Kim *et al.* [16] improved accuracy by increasing the depth to 20 layers and learning only the residuals between the high resolution image and an interpolated low resolution image. Most recently, SRResNet [18] uses many ResNet blocks to achieve state of the art pSNR and SSIM on standard super resolution benchmarks–we employ a similar design for our conditional network and catchall regression baseline.

Instead of using a per-pixel loss, Johnson *et al.*[14] use Euclidean distance between activations of a pre-trained CNN for model's predictions *vs.* ground truth images. Using this so-called preceptual loss, they train feed-forward networks for super resolution and style transfer. Bruna *et al.* [3] also use perceptual loss to train a super resolution network, but inference is done via gradient propagation to the low-res input (*e.g.,* [9]).

Ledig *et al.* [18] and Yu *et al.* [33] use GANs to create compelling super resolution results showing the ability of the model to predict plausible high frequency details. Sønderby *et al.* [15] also investigate GANs for super resolution using a learned affine transformation that ensures the models only generate images that downscale back to the low resolution inputs. Sønderby *et al.* [15] also explore a masked autoregressive model like PixelCNN [27] but without the gated layers and using a mixture of gaussians instead of a multinomial distribution. Denton *et al.* [4] use a multi-scale adversarial network for image synthesis, but the architecture also seems beneficial for super resolution.

PixelRNN and PixelCNN by Oord *et al.* [27, 28] are probabilistic generative models that impose an order on image pixels representing them as a long sequence. The probability of each pixel is then conditioned on the previous ones. The gated PixelCNN obtained state of the art log-likelihood scores on CIFAR-10 and MNIST, making it one of the most competetive probabilistic generative models.

Since PixelCNN uses log-likelihood for training, the model is highly penelized if negligible probability is assigned to any of the training examples. By contrast, GANs only learn enough to fool a non-stationary discriminator. One of their common failure cases is mode collapsing were samples are not diverse enough [21]. Furthermore, GANs require careful tuning of hyperparameters to ensure the discriminator and generator are equally powerful and learn at equal rates. PixelCNNs are more robust to hyperparameter changes and usually have a nicely decaying loss curve. Thus, we adopt PixelCNN for super resolution applications.

3. Probabilistic super resolution

We aim to learn a probabilistic super resolution model that discerns the statistical dependencies between a high resolution image and a corresponding low resolution image. Let $\mathbf{x}$ and $\mathbf{y}$ denote a low resolution and a high resolution image, where $\mathbf{y}^*$ represents a ground-truth high resolution image. In order to learn a parametric model of $p_\theta(\mathbf{y} \mid \mathbf{x})$, we exploit a large dataset of pairs of low resolution inputs and ground-truth high resolution outputs, denoted $\mathcal{D} \equiv \{(\mathbf{x}^{(i)}, \mathbf{y}^{*(i)})\}_{i=1}^N$. One can easily collect such a large dataset by starting from a set of high resolution images and lowering their resolution as much as needed. To optimize the parameters $\boldsymbol{\theta}$ of the conditional distribution p, we maximize a conditional log-likelihood objective defined as,

$$O(\boldsymbol{\theta} \mid \mathcal{D}) = \sum_{(\mathbf{x},\mathbf{y}^*)\in\mathcal{D}} \log p(\mathbf{y}^* \mid \mathbf{x}) \,. \tag{1}$$

The key problem discussed in this paper is the exact form of $p(\mathbf{y} \mid \mathbf{x})$ that enables efficient learning and inference, while generating realistic non-blurry outputs. We first discuss pixel-independent models that assume that each output pixel is generated with an independent stochastic process given the input. We elaborate why these techniques result in sub-optimal blurry super resolution results. Finally we describe our pixel recursive super resolution model that generates output pixels one at a time to enable modeling the statistical dependencies between the output pixels using PixelCNN [27, 28], and synthesizes sharp images from very blurry input.

3.1. Pixel independent super resolution

The simplest form of a probabilistic super resolution model assumes that the output pixels are conditionally independence given the inputs. As such, the conditional distribution of $p(\mathbf{y} \mid \mathbf{x})$ factorizes into a product of independent pixel predictions. Suppose an RGB output $\mathbf{y}$ has M pixels each with three color channels, *i.e.,* $\mathbf{y} \in \mathbb{R}^{3M}$. Then,

$$\log p(\mathbf{y} \mid \mathbf{x}) = \sum_{i=1}^{3M} \log p(y_i \mid \mathbf{x}) \,. \tag{2}$$

Two general forms of pixel prediction models have been explored in the literature: *Gaussian* and *multinomial* distributions to model continuous and discrete pixel values respectively. In the Gaussian case,

$$\log p(y_i \mid \mathbf{x}) = -\frac{1}{2\delta^2} \left\| y_i - C_i(\mathbf{x}) \right\|_2^2 - \log \sqrt{2\delta^2\pi} \,, \tag{3}$$

where $C_i(\mathbf{x})$ denotes the i^{th} element of a non-linear transformation of $\mathbf{x}$ via a convolutional neural network. $C_i(\mathbf{x})$ is the estimated mean for the i^{th} output pixel y_i, and σ^2 denotes the variance. Often the variance is not learned, in which case maximizing the conditional log-likelihood of (1) reduces to minimizing the mean squared error (MSE) between y_i and $C_i(\mathbf{x})$ across the pixels and channels throughout the dataset. Super resolution models based on MSE regression (*e.g.,* [5, 16, 18]) fall within this family of pixel independent models, where the outputs of a neural network parameterize a set of Gaussians with fixed bandwidth.

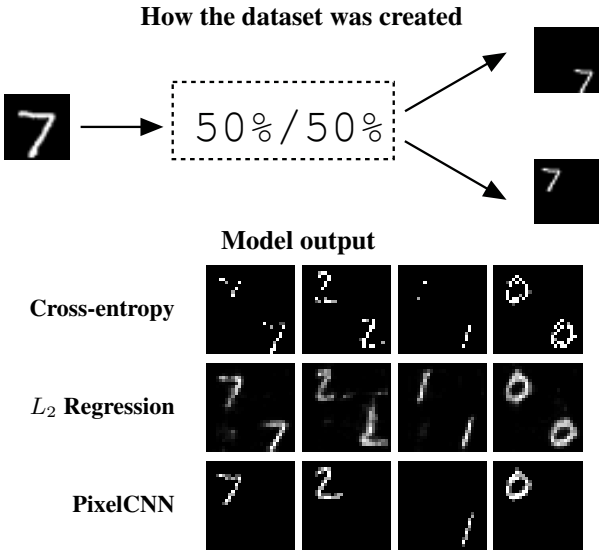


Figure 2: Top: A cartoon of how the input and output pairs for the toy dataset were created. Bottom: Example predictions for various algorithms trained on this dataset. The pixel independent L_2 regression and cross-entropy models do not exhibit multimodal predictions. The PixelCNN output is stochastic and multiple samples will place a digit in either corner 50% of the time.

Alternatively, one could use a flexible multinomial distribution as the pixel prediction model, in which case the ouput dimensions are discretized into K possible values (*e.g.,* $K = 256$) where $y_i \in \{1, \ldots, K\}$. The pixel prediction model based on a multinomial softmax operator is represented as,

$$\log p(y_i = k \mid \mathbf{x}) = \mathbf{w}_{jk}{}^{\mathsf{T}} C_i(\mathbf{x}) - \log \sum_{v=1}^{K} \exp\{\mathbf{w}_{jv}{}^{\mathsf{T}} C_i(\mathbf{x})\} \,, \tag{4}$$

where $\{\mathbf{w}_{jk}\}_{j=1,k=1}^{3,K}$ denote the softmax weights for different color channels and different discrete values.

3.2. Synthetic multimodal task

To demonstrate how the above pixel independent models can fail at conditional image modeling, we created a synthetic dataset that is explicitly multimodal. For many generative tasks like super resolution, colorization, and depth estimation, models that are able to predict a mode without averaging effects are desirable. For example, in coloriza-

tion, selecting a strong red or blue for a car is better than selecting a sepia toned average of all of the colors of cars that the model has been exposed to. In this synthetic task, the input is an MNIST digit (1st row of Figure 2), and the output is the same input digit but scaled and translated either into the upper left corner or upper right corner (2nd and 3rd rows of Figure 2). The dataset has an equal ratio of upper left and upper right outputs, which we call the MNIST corners dataset.

A convolutional network using per pixel squared error loss (Figure 2, L_2 Regression) produces two blurry figures. Replacing the continuous loss with a per-pixel cross-entropy produces crisper images but also fails to capture the stochastic bimodality because both digits are shown in both corners (Figure 2, cross-entropy). In contrast, a model that explicitly deals with multi-modality, PixelCNN stochastically predicts a digit in the upper-left or bottom-right corners but never predicts both digits simultaneously (Figure 2, PixelCNN).

See Figure 5 for examples of our super resolution model predicting different modes on a realistic dataset.

Any good generative model should be able to make sharp single mode predictions and a dataset like this would be a good starting point for any new models.

4. Pixel recursive super resolution

The main issue with the previous probabilistic models (Equations (3) and (4)) for super resolution is the lack of conditional dependency between super resolution pixels. There are two general methods to model statistical correlations between output pixels. One approach is to define the conditional distribution of the output pixels jointly by either a multivariate Gaussian mixture [36] or an undirected graphical model such as conditional random fields [8]. With these approaches one has to commit to a particular form of statistical dependency between the output pixels, for which inference can be computationally expensive. The second approach that we follow in this work, is to factorize the conditional distribution using chain rule as,

$$\log p(\mathbf{y} \mid \mathbf{x}) = \sum_{i=1}^{M} \log p(\mathbf{y}_i \mid \mathbf{x}, \mathbf{y}_{<i}) \,, \tag{5}$$

where the generation of each output dimension is conditioned on the input, previous output pixels, and the previous channels of the same output pixel. For simplicity of exposition, we ignore different output channels in our derivations, and use $\mathbf{y}_{<i}$ to represent $\{\mathbf{y}_1, \ldots, \mathbf{y}_{i-1}\}$. The benefits of this approach is that the exact form of the conditional dependencies is flexible and the inference is straightforward. Inspired by the PixelCNN model, we use a multinomial distribution to model discrete pixel values in Eq. (5). Alternatively, one could use an autoregressive prediction model with Gaussian or Logistic (mixture) conditionals as proposed in [24].

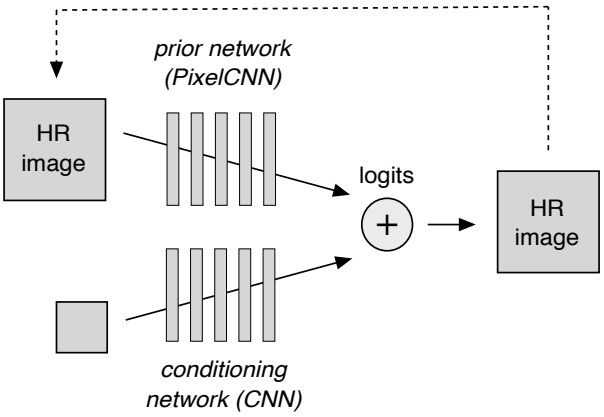


Figure 3: The proposed super resolution network comprises a *conditioning network* and a *prior network*. The *conditioning network* is a CNN that receives a low resolution image as input and outputs logits predicting the conditional log-probability of each high resolution (HR) image pixel. The *prior network*, a PixelCNN [28], makes predictions based on previous stochastic predictions (indicated by dashed line). The model's probability distribution is computed as a softmax operator on top of the sum of the two sets of logits from the prior and conditioning networks.

Our model, outlined in Figure 3, comprises two major components that are fused together at a late stage and trained jointly: (1) a *conditioning* network (2) a *prior* network. The conditioning network is a pixel independent prediction model that maps a low resolution image to a probabilistic skeleton of a high resolution image, while the prior network is supposed to add natural high resolution details to make the outputs look more realistic.

Given an input $\mathbf{x} \in \mathbb{R}^L$, let $A_i(\mathbf{x}) : \mathbb{R}^L \to \mathbb{R}^K$ denote a conditioning network predicting a vector of logit values corresponding to the K possible values that the i^{th} output pixel can take. Similarly, let $B_i(\mathbf{y}_{<i}) : \mathbb{R}^{i-1} \to \mathbb{R}^K$ denote a prior network predicting a vector of logit values for the i^{th} output pixel. Our probabilistic model predicts a distribution over the i^{th} output pixel by simply adding the two sets of logits and applying a softmax operator on them,

$$p(y_i \mid \mathbf{x}, \mathbf{y}_{<i}) = \text{softmax}(A_i(\mathbf{x}) + B_i(\mathbf{y}_{<i})) \,. \tag{6}$$

To optimize the parameters of A and B jointly, we perform stochastic gradient ascent to maximize the conditional log likelihood in (1). That is, we optimize a cross-entropy loss between the model's predictions in (6) and discrete

ground truth labels $y_i^* \in \{1, \ldots, K\}$,

$$O_1 = \sum_{(\mathbf{x},\mathbf{y}^*)\in\mathcal{D}} \sum_{i=1}^{M} \Big(\mathbb{1}[\mathbf{y}_i^*]^\mathsf{T}(A_i(\mathbf{x}) + B_i(\mathbf{y}_{<i}^*)) - lse(A_i(\mathbf{x}) + B_i(\mathbf{y}_{<i}^*)) \Big) , \tag{7}$$

where $lse(\cdot)$ is the log-sum-exp operator corresponding to the log of the denominator of a softmax, and $\mathbb{1}[k]$ denotes a K-dimensional one-hot indicator vector with its k^{th} dimension set to 1.

Our preliminery experiments indicate that models trained with (7) tend to ignore the conditioning network as the statistical correlation between a pixel and previous high resolution pixels is stronger than its correlation with low resolution inputs. To mitigate this issue, we include an additional loss in our objective to enforce the conditioning network to be optimized. This additional loss measures the cross-entropy between the conditioning network's predictions via $\mathrm{softmax}(A_i(\mathbf{x}))$ and ground truth labels. The total loss that is optimized in our experiments is a sum of two cross-entropy losses formulated as,

$$O_2 = \sum_{(\mathbf{x},\mathbf{y}^*)\in\mathcal{D}} \sum_{i=1}^{M} \Big(\mathbb{1}[\mathbf{y}_i^*]^\mathsf{T}(2\,A_i(\mathbf{x}) + B_i(\mathbf{y}_{<i}^*)) - lse(A_i(\mathbf{x}) + B_i(\mathbf{y}_{<i}^*)) - lse(A_i(\mathbf{x})) \Big) . \tag{8}$$

Once the network is trained, sampling from the model is straightforward. Using (6), starting at $i = 1$, first we sample a high resolution pixel. Then, we proceed pixel by pixel, feeding in the previously sampled pixel values back into the network, and draw new high resolution pixels. The three channels of each pixel are generated sequentially in turn.

We additionally consider *greedy decoding*, where one always selects the pixel value with the largest probability and sampling from a tempered softmax, where the concentration of a distribution p is adjusted by using a temperature parameter $\tau > 0$,

$$p_\tau = \frac{p^\tau}{\|p^\tau\|_1} .$$

To control the concentration of our sampling distribution $p(y_i \mid \mathbf{x}, \mathbf{y}_{<i})$, it suffices to multiply the logits from A and B by a parameter τ. Note that as τ goes towards ∞, the distribution converges to the mode[1], and sampling converges to greedy decoding.

4.1. Implementation details

The conditioning network is a feed-forward convolutional neural network that takes an 8×8 RGB image through a series of ResNet [10] blocks and transpose convolution layers while maintaining 32 channels throughout. The last layer uses a 1×1 convolution to increase the channels to 256×3 and uses the resulting activations to predict a multinomial distribution over 256 possible sub-pixel values via a softmax operator.

This network provides the ability to absorb the global structure of the image in the marginal probability distribution of the pixels. Due to the softmax layer it can capture the rich intricacies of the high resolution distribution, but we have no way to coherently sample from it. Sampling sub-pixels independently will mix the assortment of distributions.

The prior network provides a way to tie together the sub-pixel distributions and allow us to take samples dependent on each other. We use 20 gated PixelCNN layers with 32 channels at each layer. We leave conditioning until the late stages of the network, where we add the pre-softmax activations from the conditioning network and prior network before computing the final joint softmax distribution.

Our model is built by using TensorFlow [1] and trained across 8 GPUs with synchronous SGD updates. See appendix A for further details.

5. Experiments

We assess the effectiveness of the proposed pixel recursive super resolution method on two datasets containing small faces and bedroom images. The first dataset is a version of the CelebA dataset [19] composed of a set of celebrity faces, which are cropped around the face. In the second dataset LSUN Bedrooms [32], images are center cropped. In both datasets we resize the images to 32×32 with bicubic interpolation and again to 8×8, constituting the output and input pairs for training and evaluation. We present representative super resolution examples on held out test sets and report human evaluations of our predictions in Table 1.

We compare our results with two baselines: a pixel independent L_2 regression ("Regression") and a nearest neighbors search ("NN"). The architecture used for the regression baseline is identical to the conditioning network used in our model, consisting of several ResNet blocks and upsampling convolutional layers, except that the baseline regression model outputs three channels and has a final $\tanh(\cdot)$ non-linearity instead of ReLU. The regression architecture is similar in design to to SRResNet [18], which reports state of the art scores in image similarity metrics. Furthermore, we train the regression network to predict super resolution residuals instead of the actual pixel values. The residuals are computed based on bicubic interpolation of the input, and are known to work better to provide superior predictions [16]. The nearest neighbors baseline searches the downsampled training set for the nearest example (using eu-

[1] We use a non-standard notion of temperature that represents $\frac{1}{\tau}$ in the standard notation.

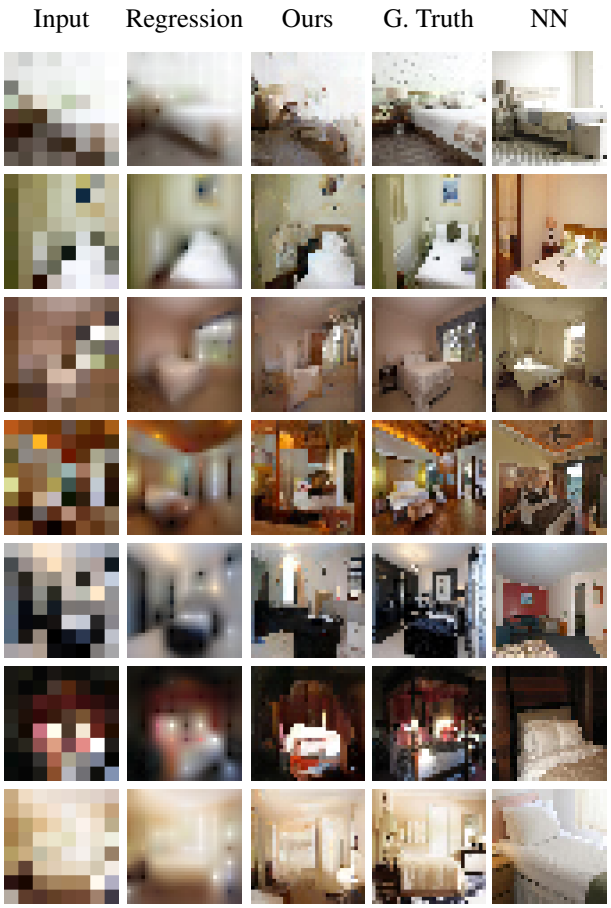


Figure 4: Samples from the model trained on LSUN Bedrooms at 32×32.

clidean distance) and returns its high resolution counterpart.

5.1. Sampling

Sampling from the model multiple times results in different high resolution images for a given low resolution image (Figure 5). A given model will identify many plausible high resolution images that correspond to a given lower resolution image. Each one of these samples may contain distinct qualitative features and each of these modes is contained within the PixelCNN. Note that the differences between samples for the faces dataset are far less drastic than seen in our synthetic dataset, where failure to cleanly predict modes meant complete failure.

The quality of samples is sensitive to the softmax temperature. When the mode is sampled ($\tau = \infty$) at each subpixel, the samples are of poor quality, they look smooth with horizontal and vertical line artifacts. Sampling at $\tau = 1.0$, the exact probability given by the model, tend to be more jittery with high frequency content. It seems in this case there are multiple less certain trajectories and the samples jump back and forth between them–perhaps this is alleviated with more capacity and training time. Manually tuning the softmax temperature was necessary to find good looking samples–usually a value between 1.1 and 1.3 worked.

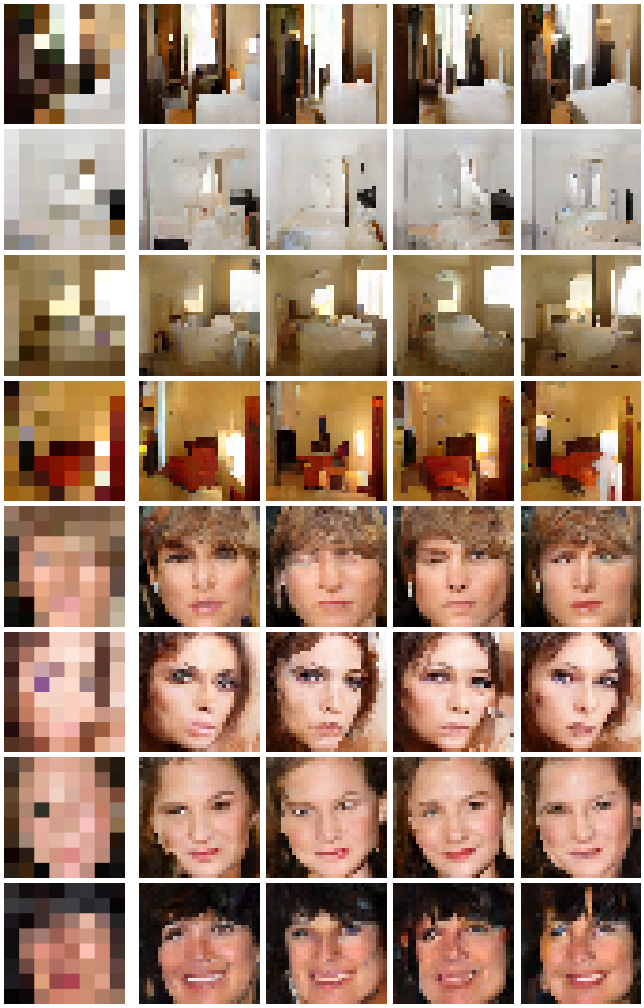


Figure 5: Left: low-res input. Right: Diversity of super resolution samples at $\tau = 1.2$.

In Figure 6 are various test predictions with their negative log probability scores listed below each image. Smaller scores means the model has assigned that image a larger probability mass. The greedy, bicubic, and regression faces are preferred by the model despite being worse quality. This is probably because their smooth face-like structure doesn't contradict the learned distributions. Yet sampling with the proper softmax temperature nevertheless finds realistic looking images.

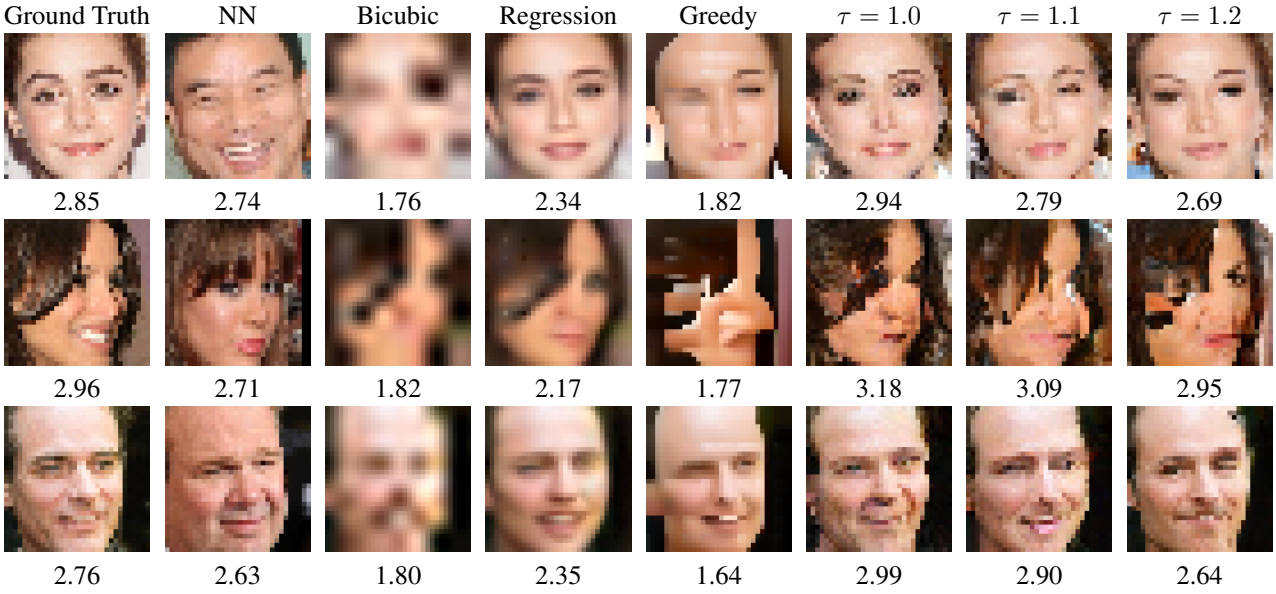


Figure 6: Our model does not produce calibrated log-probabilities for the samples. Negative log-probabilities are reported below each image. Note that the best log-probabilities arise from bicubic interpolation and greedy sampling even though the images are poor quality.

5.2. Image similarity

Many methods exist for quantifying image similarity that attempt to measure human perception judgements of similarity [29, 30, 20]. We quantified the prediction accuracy of our model compared to ground truth using pSNR and MS-SSIM (Table 1). We found that our own visual assessment of the predicted image quality did not correspond to these image similarities metrics. For instance, bicubic interpolation achieved relatively high metrics even though the samples appeared quite poor. This result matches recent observations that suggest that pSNR and SSIM provide poor judgements of super resolution quality [18, 14] when new details are synthesized.

To ensure that samples do indeed correspond to the low-resolution input, we measured how consistent the high resolution output image is with the low resolution input image (Table 1, 'consistency'). Specifically, we measured the L2 distance between the low-resolution input image and a bicubic downsampled version of the high resolution estimate. Lower L2 distances correspond to high resolutions that are more similar to the original low resolution image. Note that the nearest neighbor high resolution images are less consistent even though we used a database of 3 million training images to search for neighbors in the case of LSUN bedrooms. In contrast, the bicubic resampling and the PixelCNN upsampling methods showed consistently better consistency with the low resolution image. This indicates that our samples do indeed correspond to the low-resolution input.

5.3. Human study

We presented crowd sourced workers with two images: a true image and the corresponding prediction from our various models. Workers were asked "Which image, would you guess, is from a camera?" Following the setup in Zhang et al [34], we present each image for one second at a time before allowing them to answer. Workers are started with 10 practice pairs during which they get feedback if they choose correctly or not. The practice pairs not counted in the results. After the practice pairs, they are shown 45 additional pairs, 5 of which are golden questions designed to test if the person is paying attention. The golden question pits a bicubicly upsampled image (very blurry) against the ground truth. Excluding the golden and practice questions, we count fourty answers per session. Sessions in which they missed any golden questions are thrown out. Workers were only allowed to participate in any of our studies once. We continued running sessions until fourty different different workers were tested on each of the four algorithms.

We report in Table 1 the percent of the time users choose an algorithm's output over the ground truth counterpart. Note that 50% would say that an algorithm perfectly confused the subjects.

Algorithm	pSNR	SSIM	MS-SSIM	Consistency	% Fooled
Bicubic	28.92	0.84	0.76	0.006	–
NN	28.18	0.73	0.66	0.024	–
Regression	**29.16**	**0.90**	**0.90**	**0.004**	4.0 ± 0.2
$\tau = 1.0$	29.09	0.84	0.86	0.008	$\mathbf{11.0 \pm 0.1}$
$\tau = 1.1$	29.08	0.84	0.85	0.008	10.4 ± 0.2
$\tau = 1.2$	29.08	0.84	0.86	0.008	10.2 ± 0.1
Bicubic	**28.94**	0.70	0.70	**0.002**	–
NN	28.15	0.49	0.45	0.040	–
Regression	28.87	**0.74**	**0.75**	0.003	2.1 ± 0.1
$\tau = 1.0$	28.92	0.58	0.60	0.016	17.7 ± 0.4
$\tau = 1.1$	28.92	0.59	0.59	0.017	22.4 ± 0.3
$\tau = 1.2$	28.93	0.59	0.58	0.018	$\mathbf{27.9 \pm 0.3}$

Table 1: Top: Results on the cropped CelebA test dataset at 32×32 magnified from 8×8. Bottom: LSUN bedrooms. pSNR, SSIM, and MS-SSIM measure image similarity between samples and the ground truth. Consistency lists the MSE between the input low-res image and downsampled samples on a $[0, 1]$ scale. % Fooled reports how often the algorithms samples fooled a human in a crowd sourced study; 50% would be perfectly confused.

6. Conclusion

As in many image transformation tasks, the central problem of super resolution is in hallucinating sharp details by choosing a mode of the output distribution. We explored this underspecified problem using small images, demonstrating that even the smallest 8×8 images can be enlarged to sharp 32×32 images. We introduced a toy dataset with a small number of explicit modes to demonstrate the failure cases of two common pixel independent likelihood models. In the presented model, the conditioning network gets us most of the way towards predicting a high-resolution image, but the outputs are blurry where the model is uncertain. Combining the conditioning network with a PixelCNN model provides a strong prior over the output pixels, allowing the model to generate crisp predications. Our human evaluations indicate that samples from our model on average look more photo realistic than a strong regression based conditioning network alone.

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org. 5

[2] M. Aharon, M. Elad, and A. Bruckstein. Svdd: An algorithm for designing overcomplete dictionaries for sparse representation. *Trans. Sig. Proc.*, 54(11):4311–4322, Nov. 2006. 2

[3] J. Bruna, P. Sprechmann, and Y. LeCun. Super-resolution with deep convolutional sufficient statistics. *CoRR*, abs/1511.05666, 2015. 2

[4] E. L. Denton, S. Chintala, A. Szlam, and R. Fergus. Deep generative image models using a laplacian pyramid of adversarial networks. *NIPS*, 2015. 2

[5] C. Dong, C. C. Loy, K. He, and X. Tang. Image super-resolution using deep convolutional networks. *CoRR*, abs/1501.00092, 2015. 2, 3

[6] R. Fattal. Image upsampling via imposed edge statistics. *ACM Trans. Graph.*, 26(3), July 2007. 2

[7] W. T. Freeman, T. R. Jones, and E. C. Pasztor. Example-based super-resolution. *IEEE Computer graphics and Applications*, 2002. 2

[8] W. T. Freeman and E. C. Pasztor. Markov networks for super-resolution. In *CISS*, 2000. 4

[9] L. A. Gatys, A. S. Ecker, and M. Bethge. A neural algorithm of artistic style. *CoRR*, abs/1508.06576, 2015. 2

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CVPR*, 2015. 2, 5

[11] H. Hou and H. Andrews. Cubic splines for image interpolation and digital filtering. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 26(6):508–517, Jan. 2003. 2

[12] J. Huang and D. Mumford. Statistics of natural images and models. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 1. IEEE, 1999. 2

[13] J.-B. Huang, A. Singh, and N. Ahuja. Single image super-resolution from transformed self-exemplars. In *IEEE Conference on Computer Vision and Pattern Recognition)*, 2015. 2

[14] J. Johnson, A. Alahi, and F. Li. Perceptual losses for real-time style transfer and super-resolution. *CoRR*, abs/1603.08155, 2016. 2, 7

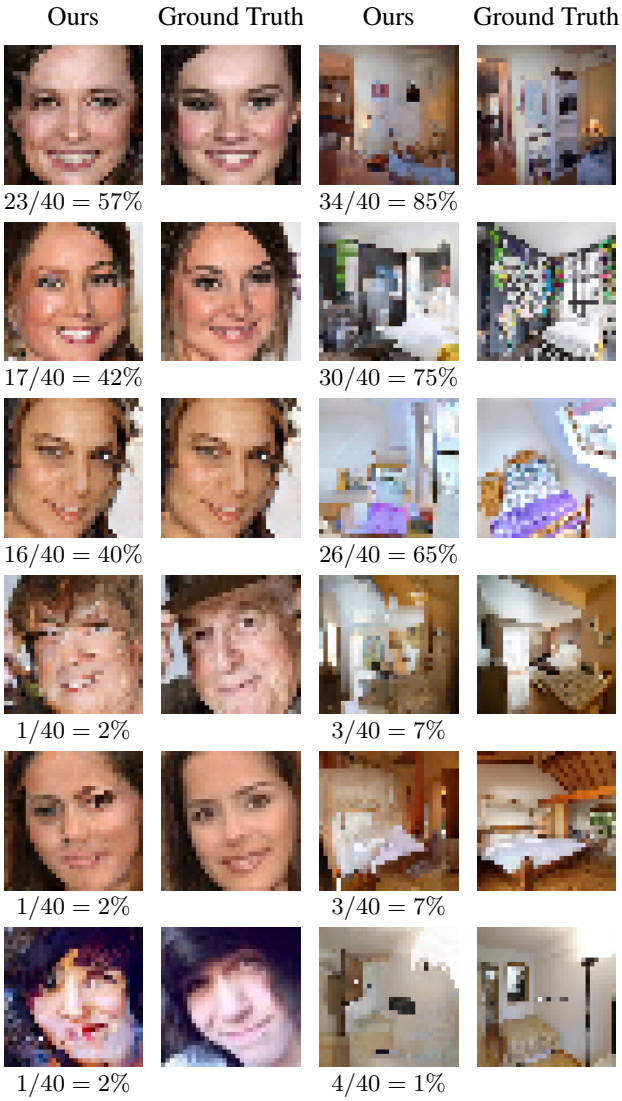


Figure 7: The best and worst rated images in the human study. The fractions below the images denote how many times a person choose that image over the ground truth. See the supplementary material for more images used in the study.

[15] C. Kaae Sønderby, J. Caballero, L. Theis, W. Shi, and F. Huszár. Amortised MAP Inference for Image Super-resolution. *ArXiv e-prints*, Oct. 2016. 2

[16] J. Kim, J. K. Lee, and K. M. Lee. Accurate image super-resolution using very deep convolutional networks. *CoRR*, abs/1511.04587, 2015. 2, 3, 5

[17] K. I. Kim and Y. Kwon. Single-image super-resolution using sparse regression and natural image prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(6):1127–1133, 2010. 2

[18] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-realistic single image super-resolution using a generative adversarial network. *arXiv:1609.04802*, 2016. 2, 3, 5, 7

[19] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015. 5

[20] K. Ma, Q. Wu, Z. Wang, Z. Duanmu, H. Yong, H. Li, and L. Zhang. Group mad competition - a new methodology to compare objective image quality models. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 6

[21] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein. Unrolled generative adversarial networks. *CoRR*, abs/1611.02163, 2016. 2

[22] K. Nasrollahi and T. B. Moeslund. Super-resolution: A comprehensive survey. *Mach. Vision Appl.*, 25(6):1423–1468, Aug. 2014. 2

[23] Y. Romano, J. Isidoro, and P. Milanfar. RAISR: rapid and accurate image super resolution. *CoRR*, abs/1606.01299, 2016. 2

[24] T. Salimans, A. Karpathy, X. Chen, D. P. Kingma, and Y. Bulatov. Pixelcnn++: A pixelcnn implementation with discretized logistic mixture likelihood and other modifications. under review at ICLR 2017. 4

[25] Q. Shan, Z. Li, J. Jia, and C.-K. Tang. Fast image/video upsampling. *ACM Transactions on Graphics (TOG)*, 27(5):153, 2008. 2

[26] J. Sun, Z. Xu, and H.-Y. Shum. Image super-resolution using gradient profile prior. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. 2

[27] A. van den Oord, N. Kalchbrenner, and K. Kavukcuoglu. Pixel recurrent neural networks. *ICML*, 2016. 2, 3

[28] A. van den Oord, N. Kalchbrenner, O. Vinyals, L. Espeholt, A. Graves, and K. Kavukcuoglu. Conditional image generation with pixelcnn decoders. *NIPS*, 2016. 2, 3, 4

[29] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 6

[30] Z. Wang, E. P. Simoncelli, and A. C. Bovik. Multiscale structural similarity for image quality assessment. In *Signals, Systems and Computers, 2004. Conference Record of the Thirty-Seventh Asilomar Conference on*, volume 2, pages 1398–1402. Ieee, 2004. 6

[31] C. Y. Yang, S. Liu, and M. H. Yang. Structured face hallucination. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1099–1106, June 2013. 2

[32] F. Yu, Y. Zhang, S. Song, A. Seff, and J. Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 5

[33] X. Yu and F. Porikli. *Ultra-Resolving Face Images by Discriminative Generative Networks*, pages 318–333. Springer International Publishing, Cham, 2016. 2

[34] R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. *ECCV*, 2016. 7

[35] D. Zoran and Y. Weiss. From learning models of natural image patches to whole image restoration. In *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, pages 479–486, Washington, DC, USA, 2011. IEEE Computer Society. 2

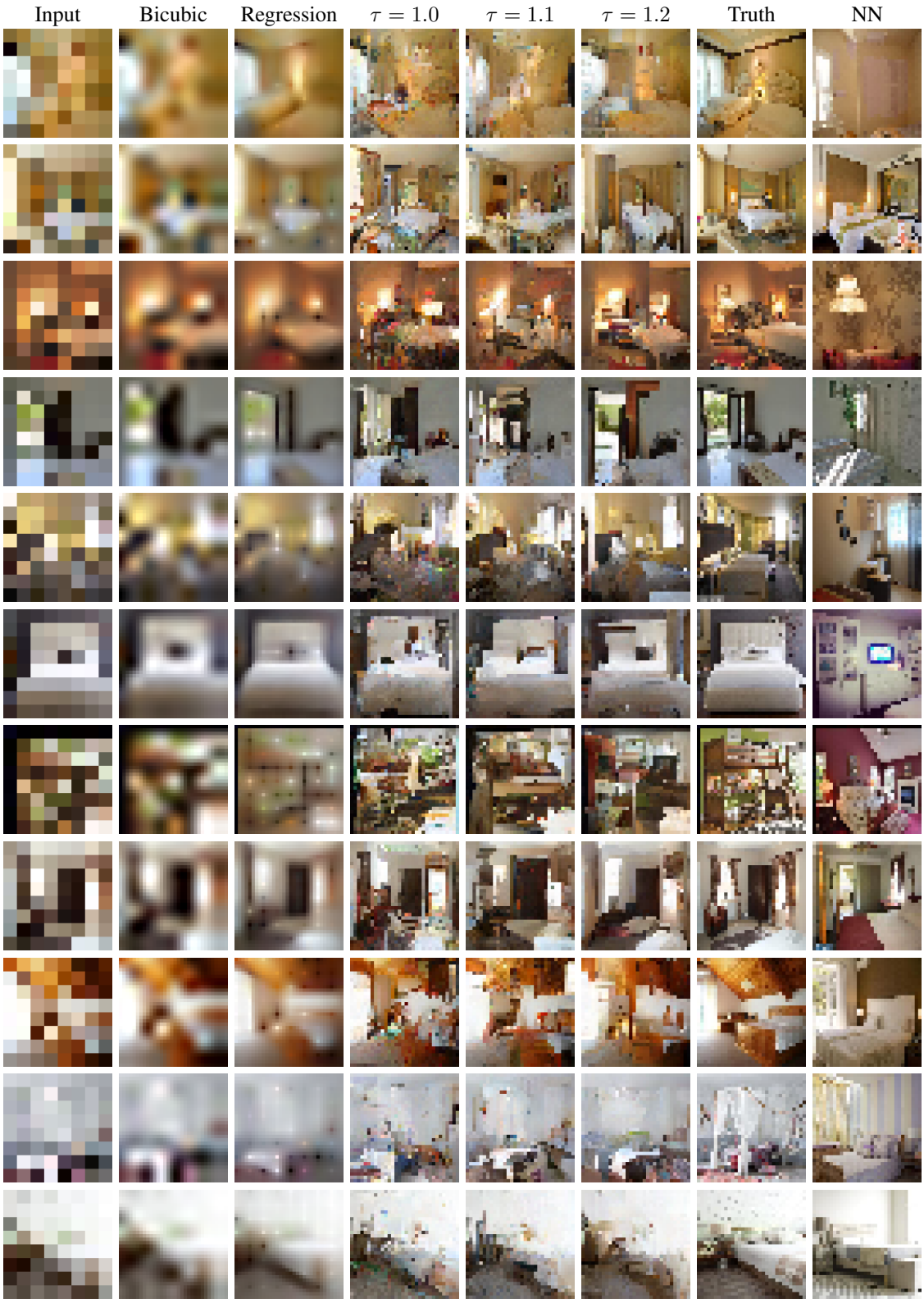
[36] D. Zoran and Y. Weiss. From learning models of natural image patches to whole image restoration. In *CVPR*, 2011. 4

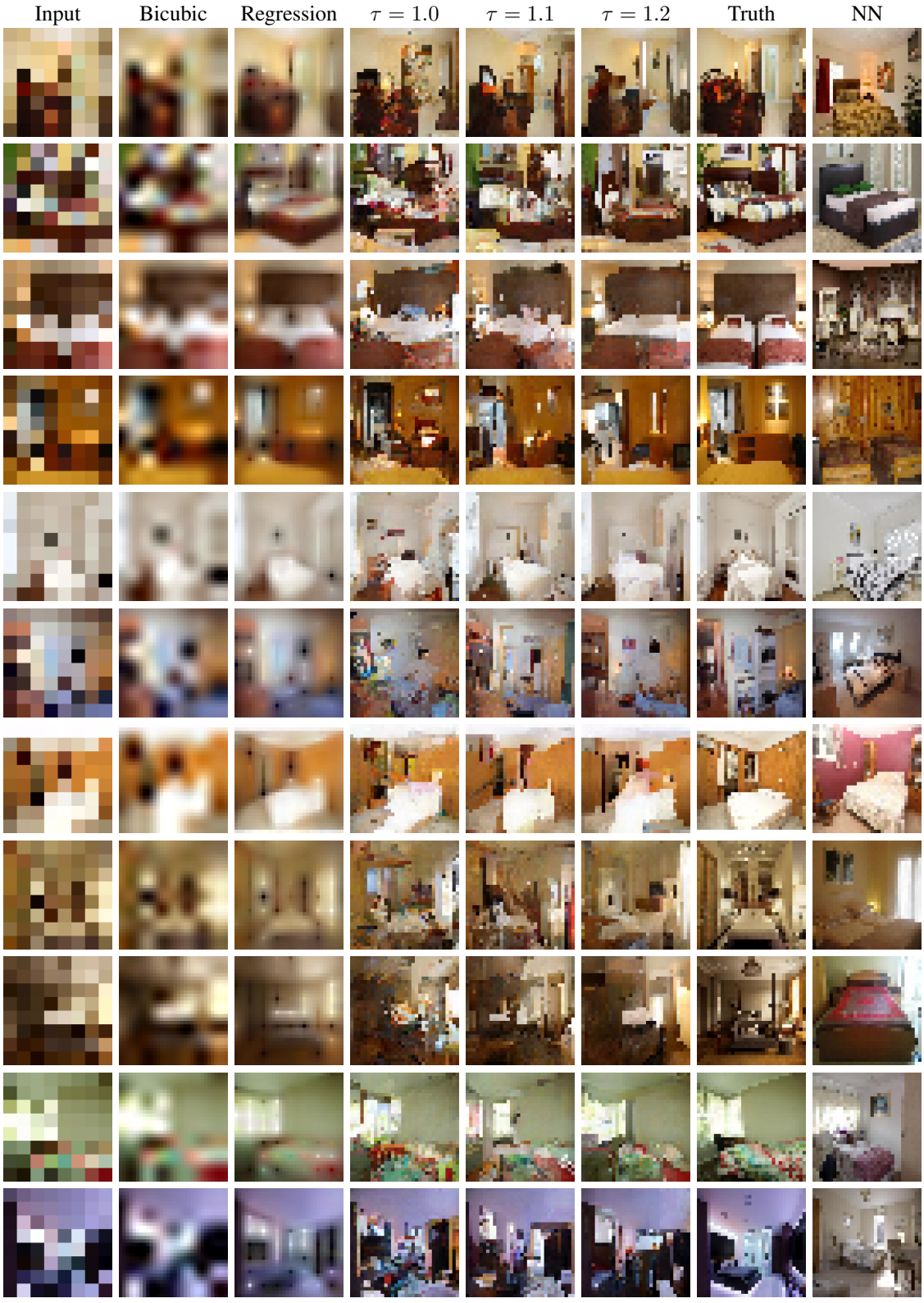
A.

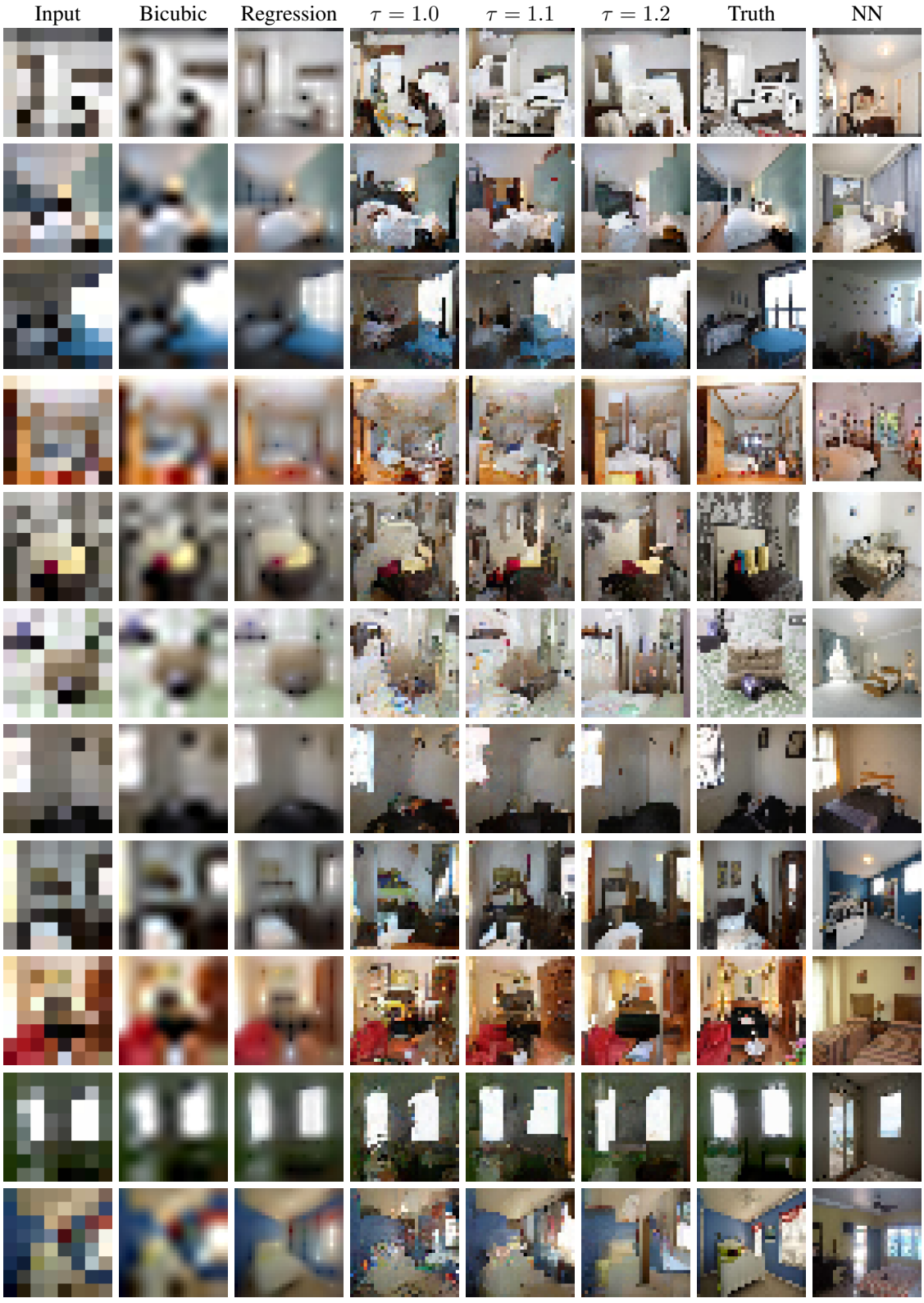
Operation	Kernel	Strides	Feature maps
Conditional network – $8 \times 8 \times 3$ input			
$B \times$ ResNet block	3×3	1	32
Transposed Convolution	3×3	2	32
$B \times$ ResNet block	3×3	1	32
Transposed Convolution	3×3	2	32
$B \times$ ResNet block	3×3	1	32
Convolution	1×1	1	$3 * 256$
PixelCNN network – $32 \times 32 \times 3$ input			
Masked Convolution	7×7	1	64
$20 \times$ Gated Convolution Layers	5×5	1	64
Masked Convolution	1×1	1	1024
Masked Convolution	1×1	1	$3 * 256$
Optimizer	RMSProp (decay=0.95, momentum=0.9, epsilon=1e-8)		
Batch size	32		
Iterations	2,000,000 for Bedrooms, 200,000 for faces.		
Learning Rate	0.0004 and divide by 2 every 500000 steps.		
Weight, bias initialization	truncated normal (stddev=0.1), Constant(0)		

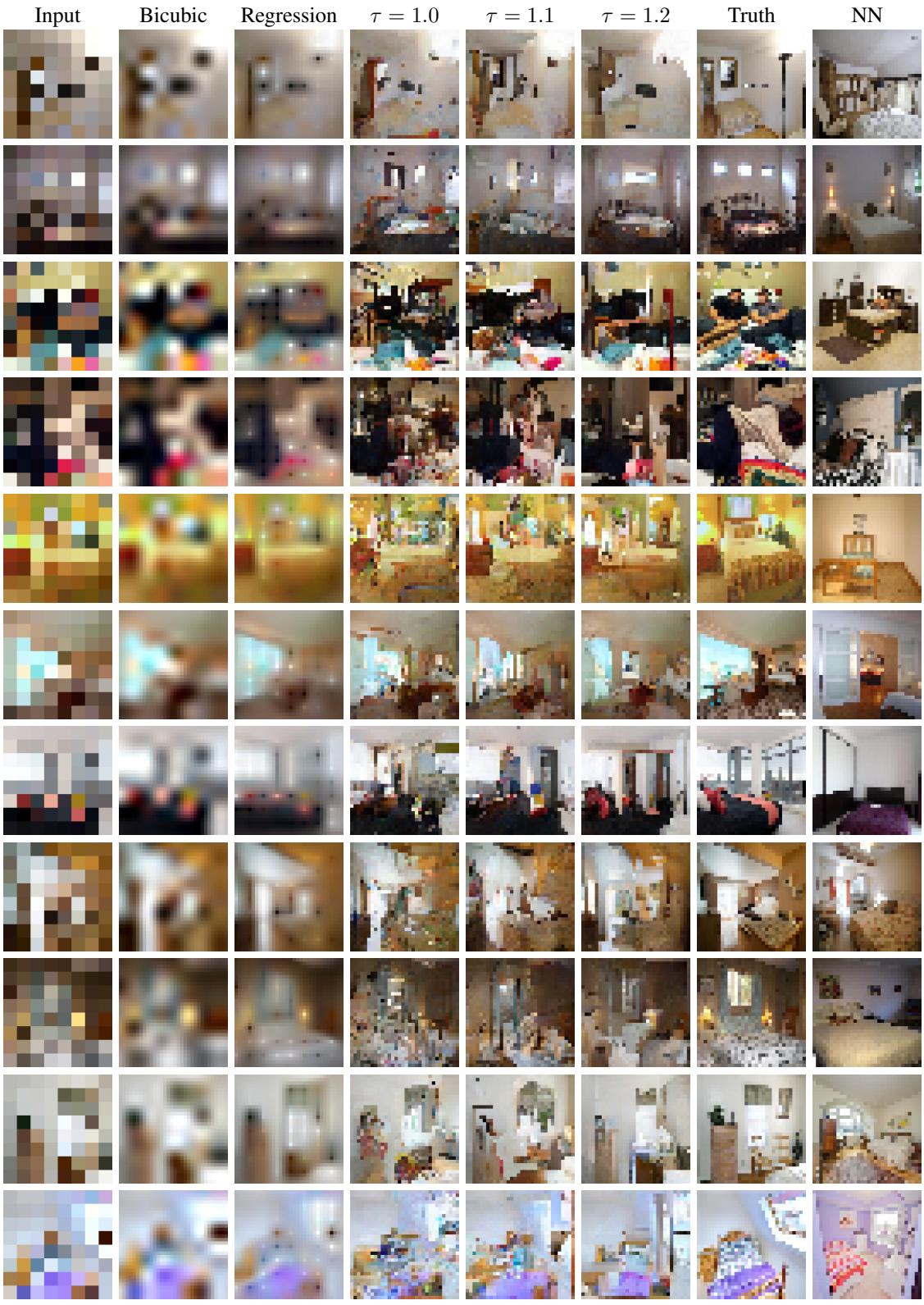
Table 2: Hyperparameters used for both datasets. For LSUN bedrooms $B = 10$ and for the cropped CelebA faces $B = 6$.

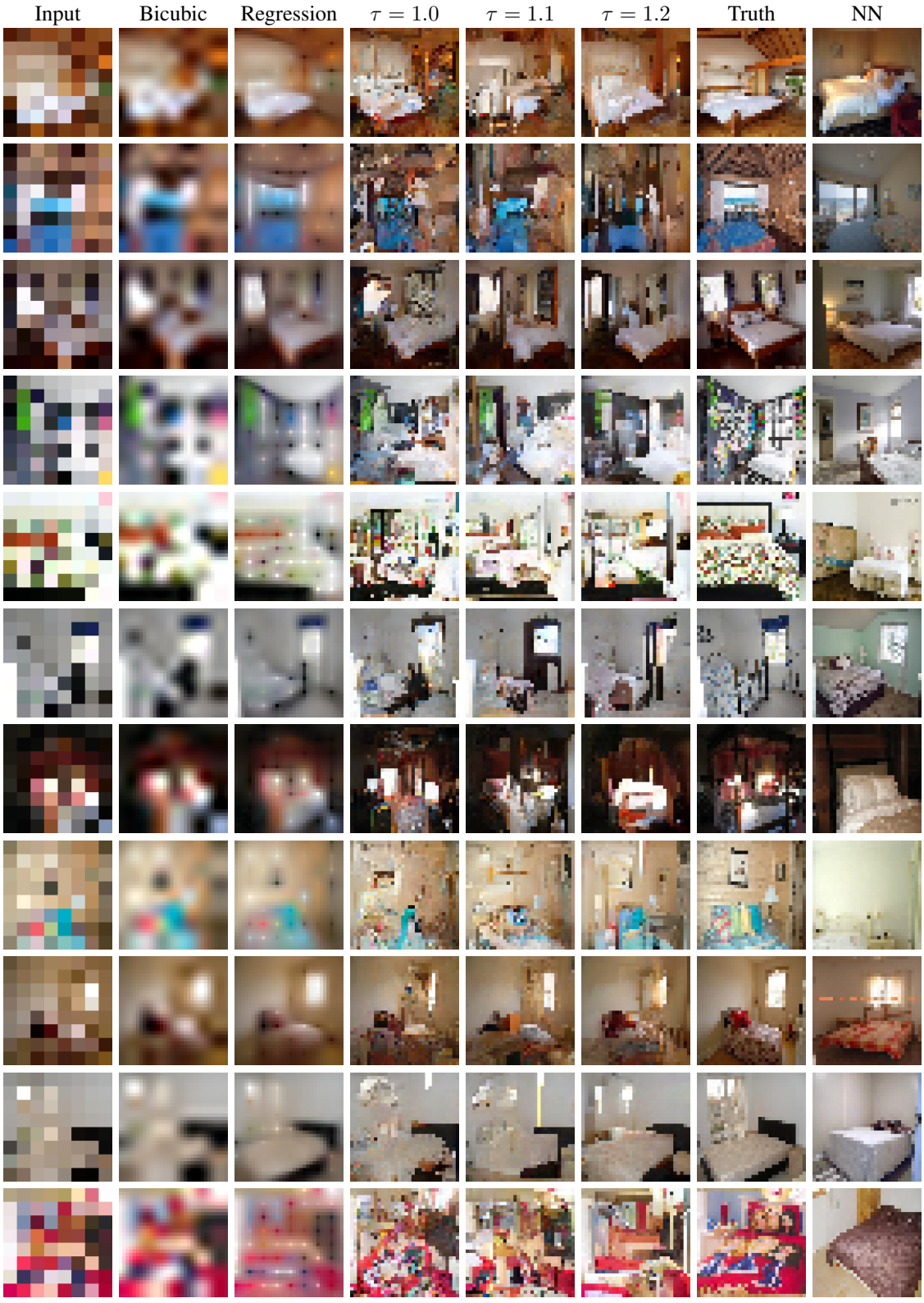
B. LSUN bedrooms samples



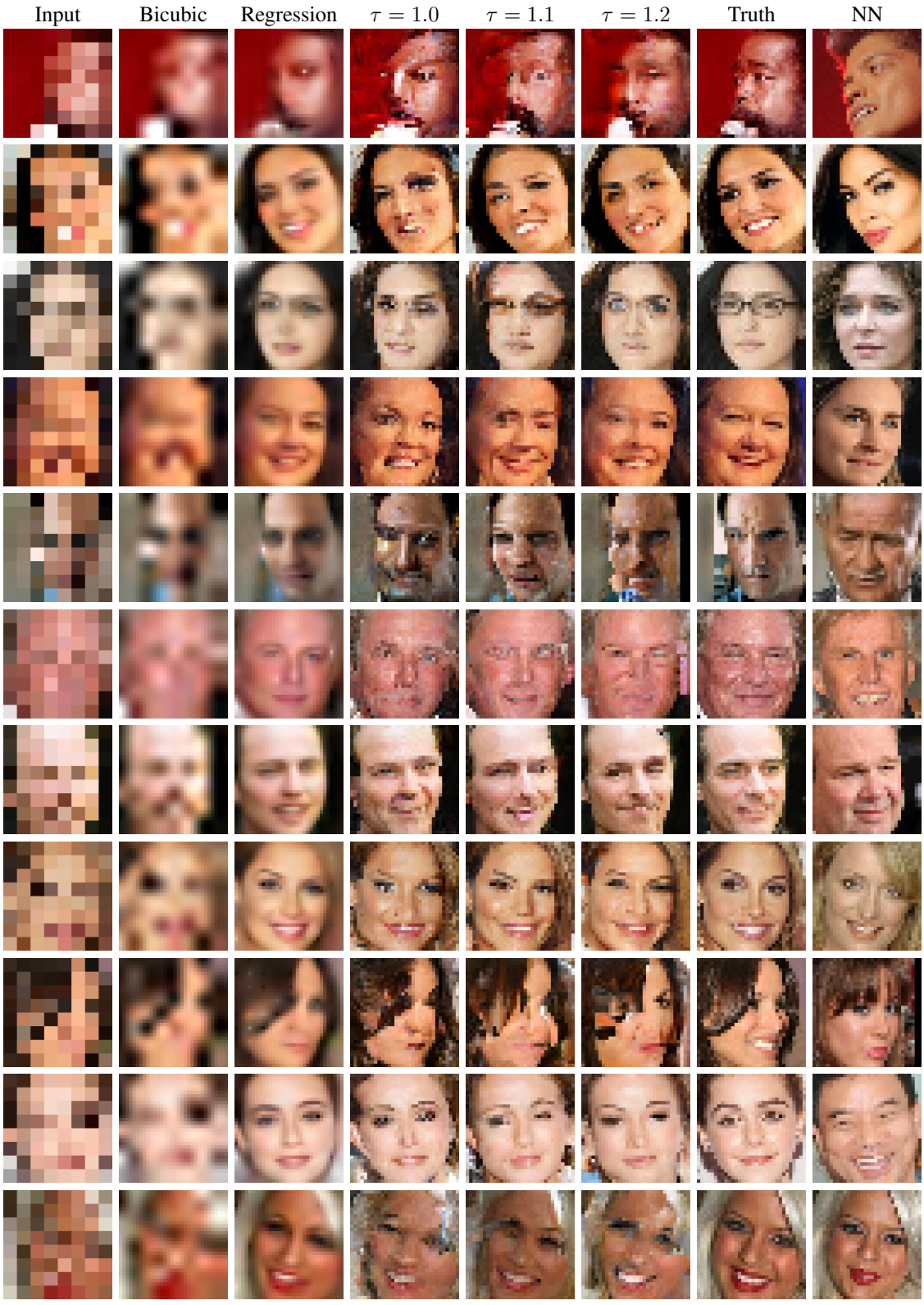

Input
Bicubic
Regression
$\tau = 1.0$
$\tau = 1.1$
$\tau = 1.2$
Truth
NN

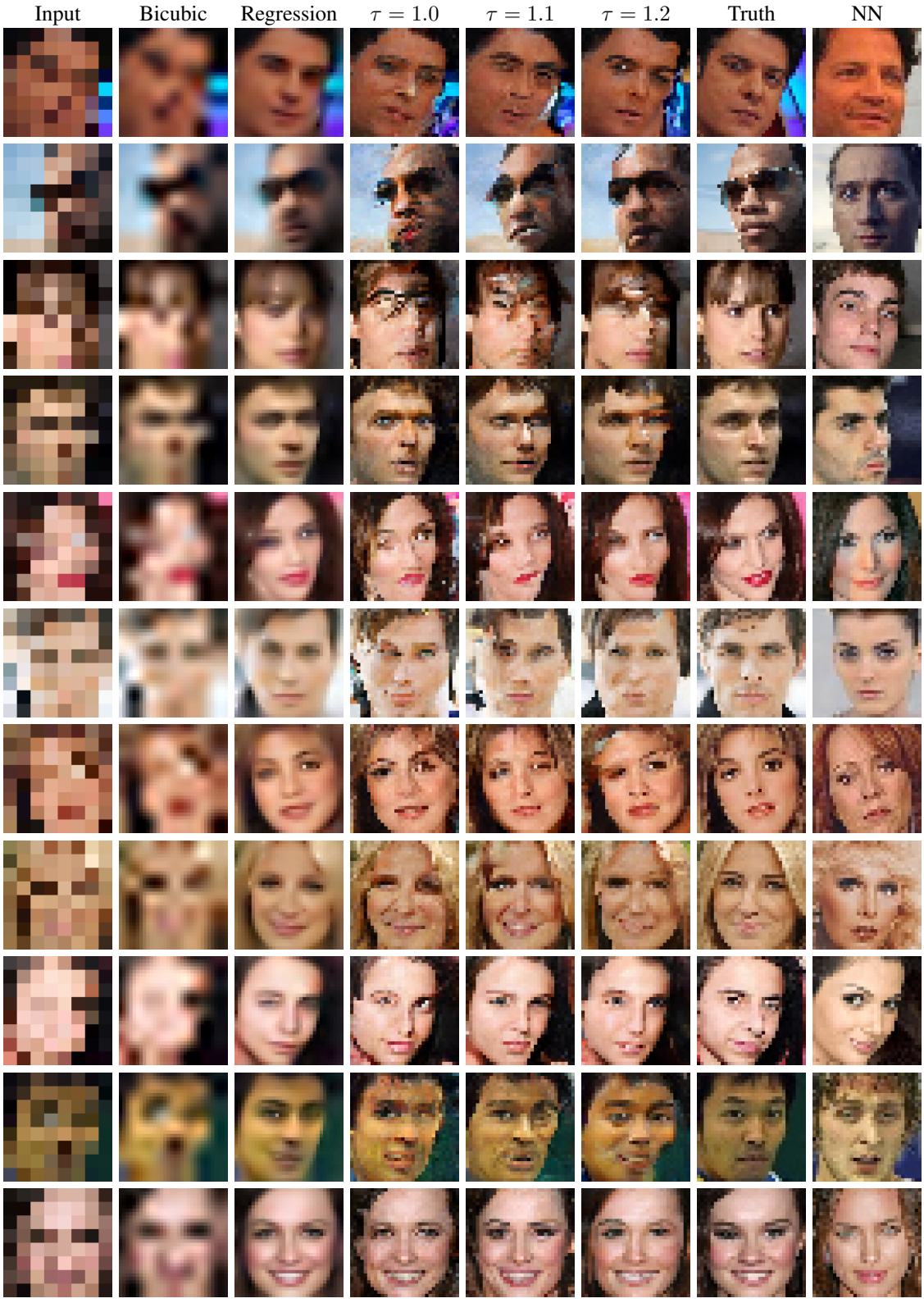

Input
Bicubic
Regression
$\tau = 1.0$
$\tau = 1.1$
$\tau = 1.2$
Truth
NN

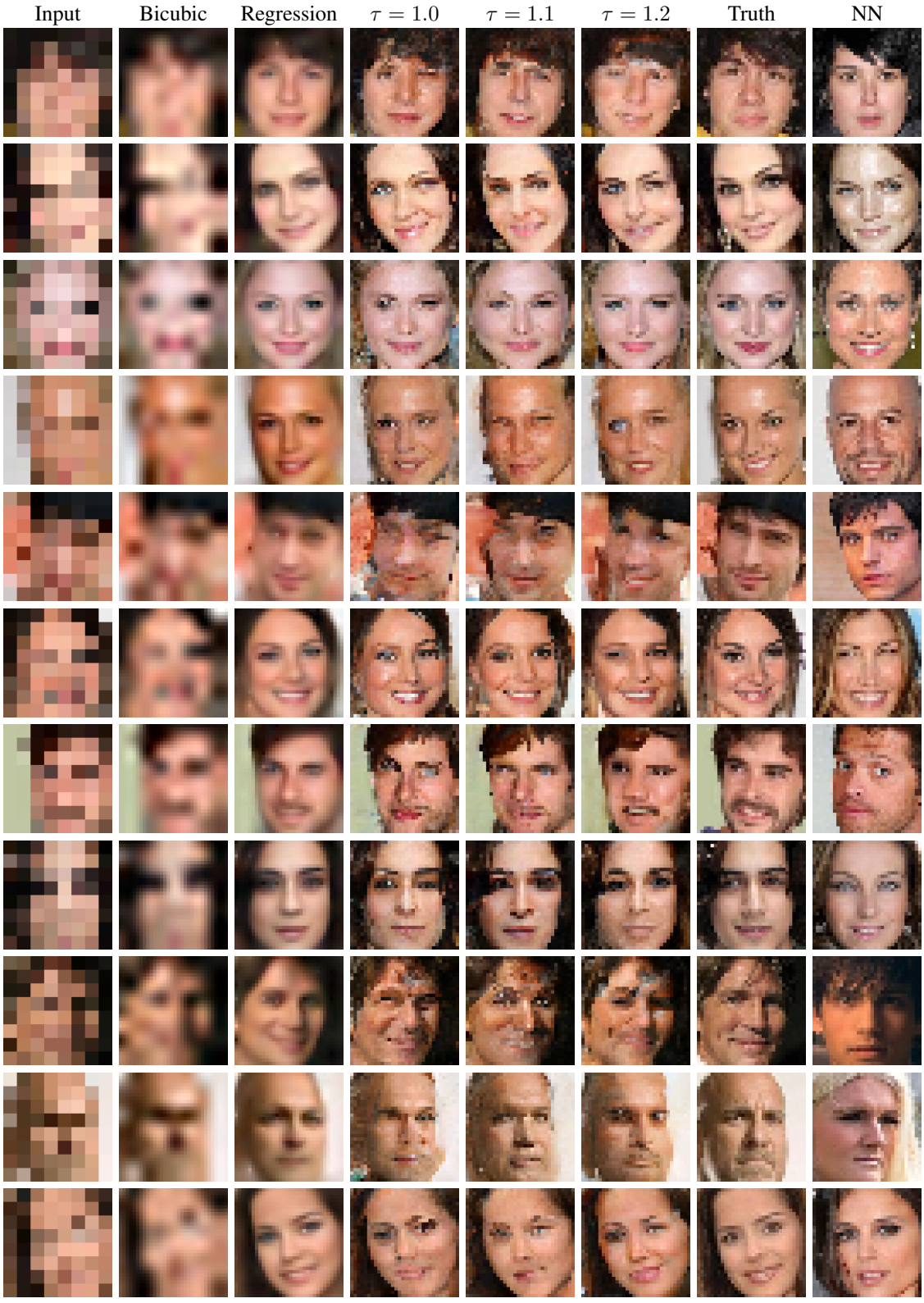

Input
Bicubic
Regression
$\tau = 1.0$
$\tau = 1.1$
$\tau = 1.2$
Truth
NN

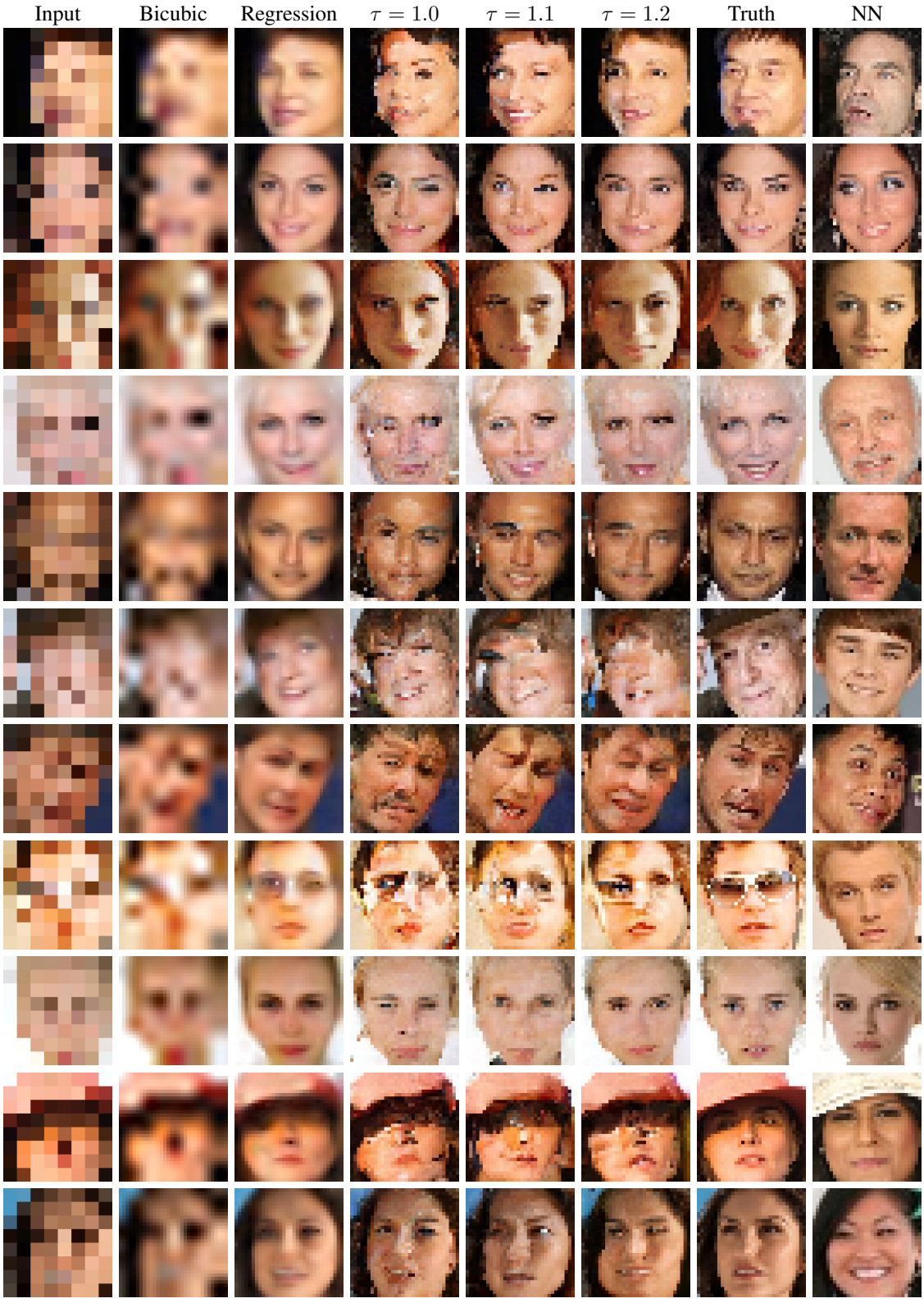

Input
Bicubic
Regression
$\tau = 1.0$
$\tau = 1.1$
$\tau = 1.2$
Truth
NN

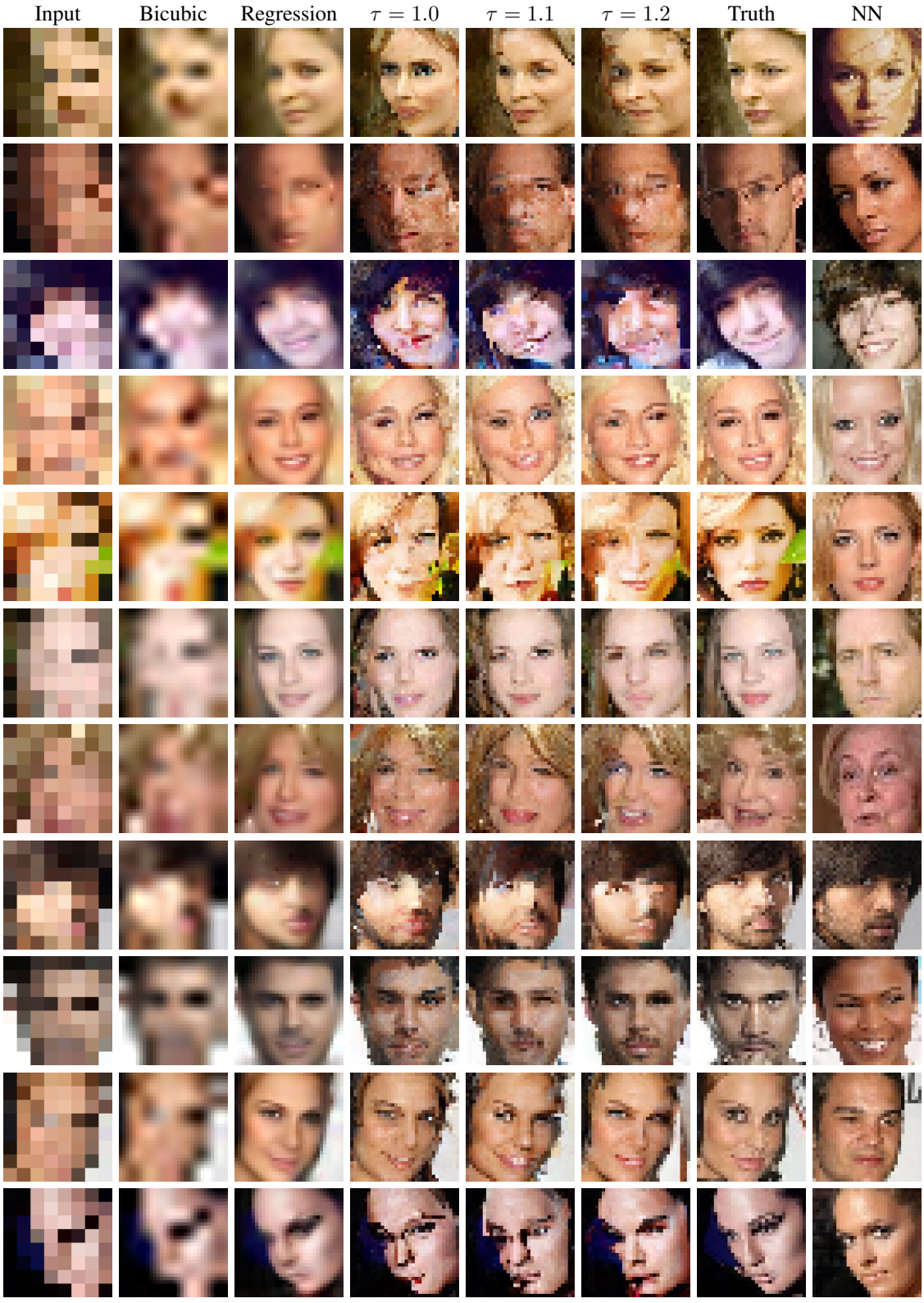
C. Cropped CelebA faces




Input
Bicubic
Regression
$\tau = 1.0$
$\tau = 1.1$
$\tau = 1.2$
Truth
NN


Input
Bicubic
Regression
$\tau = 1.0$
$\tau = 1.1$
$\tau = 1.2$
Truth
NN


Input
Bicubic
Regression
$\tau = 1.0$
$\tau = 1.1$
$\tau = 1.2$
Truth
NN


Input
Bicubic
Regression
$\tau = 1.0$
$\tau = 1.1$
$\tau = 1.2$
Truth
NN