COURSE PROJECT:
IMAGE SUPER RESOLUTION WITH DEEP LEARNING AND MODERN METHODS

Colin Greeley
CptS 580: Computer Vision
Department of Computer Science
Washington State University
Pullman, Washington
colin.greeley@wsu.edu

1 INTRODUCTION

Image super-resolution is the process of mapping an image of low-resolution (LR) to high-resolution (HR) while maintaining or increasing the feature space of the image. This is a challenging task that has been popular in the field if computer vision and deep learning since deep convolutional neural networks have shown to have success in learning how to preserve or create features of an image during image scaling. Image re-scaling has been done using standard interpolation techniques built from statistical and geometric properties of 2-dimensional images and linear transformations. The SRGAN (Ledig et al. (2016)) model has become the standard baseline for deep learning super-resolution methods due to its superior ability to increase the resolution of an image while also create features that are almost indistinguishably photo-realistic. The authors of SRGAN (Ledig et al. (2016)) introduces many clever techniques that resulted in the generated image having far higher quality than those generated from state-of-the-art deep learning methods. The deep learning model proposed in this paper is an extension of the original SRGAN by implementing more modern deep learning techniques to allow for high quility image understanding from the model leading to high quality image generation.

2 RELATED WORK

2.1 IMAGE SUPER-RESOLUTION

The first method of image super sampling dates back to the beginning of digital images themselves, as resizing an image from LR to HR via grid interpolation is the first form of image super sampling. There are a handful of filters used to interpolate a digital image to another resolution, some of which are nearest-neighbor, bilinear, or bicubic interpolation. While these interpolation methods are very fast and cheap to compute, they often results in a smudged or blurry looking image when used for super sampling since they are not actually creating information, they are just used to interpolate information between the known grid (pixel) values.

Deep learning algorithms have become the most successful for image super-sampling in in terms of predicting an HR image that is the most indistinguishable from the ground truth HR image. Sparse image coding along with a deep feed forward neural network was among the first deep learning methods for super resolution (Wang et al. (2015)). Another successful deep learning super-resolution method is to learn a short convolutional encoding between a up-sampled LR image via bicubic interpolation and the ground truth HR image (Dong et al. (2016)). The most popular method which is the baseline for many state-of-the-art solutions is the super-resolution generative adversarial network (SRGAN) (Ledig et al. (2016)). SRGAN is a deep convolutional neural network (CNN) that maps LR images to HR images using learned convolutional features. Most of the contribution in the paper is in the loss functions which are key to successful training. The first loss function used for the image super resolution CNN model is MSE loss, defined as the mean squared error between the pixel values in the predicted image and the ground truth HR image. The second loss function is the content loss, defined as the euclidean distance between feature

vectors of a pre-trained model generated by feeding in both the predicted and HR images to the model and taking the difference. In the case of SRGAN, the pretrained model is VGG19 (Simonyan & Zisserman (2015)) where the output feature vector is the i-th convolutional layer within the network. This loss function allows the super resolution CNN to be able to learn what features of an image to learn as well as the correct pixel values. The final loss function that has pushed the SRGAN to become overwhelmingly more successful that its predecessors is generative adversarial loss from a discriminator network.

2.2 Convolutional neural networks and deep learning

Convolutional neural networks (CNNs) have shown to be the superior method in solving almost all image-processing related tasks since the beginning of the artificial intelligence revolution in the 2010's (O'Shea & Nash (2015)). The more complex a problem is, generally the more parameters a CNN needs to be sucessful in solving such a problem. Deep CNNs became possible and easily trainable with the discovery of residual connections shown in ResNet (He et al. (2015)). The current state-of-the-art super-resolution CNN would not be possible without the help of residual connections as these models tend to be very deep, i.e., 100+ convolutional layers.

2.3 Generative adversarial networks

Generative adversarial networks (GANs) (Goodfellow et al. (2014)) have been one of the most successful methods for training a deep learning model to generate realistic looking images. The full system consists of two CNNs, the fist is a decoder model called the generator which maps an input feature vector of noise to a generated image. The second model is a standard CNN image classification model which maps input images to a validity score $[0, 1)$ which represents if the input image is real or fake. Training a GAN is a back-and-forth battle between the generator and discriminator since the discriminators goal is to be able to classify if an image is real or fake while the generator's goal is to trick the discriminator into thinking the the fake image is real. The generator is trained through sparse gradients passed back though the discriminator and convergence is reached when the discriminator can no longer distinguish weather an image is real or fake.

There have been many iterations of improvement since the invention of the first GAN (Goodfellow et al. (2014)). The most notable in relation to this project are the DCGAN (Radford et al. (2015)) and ACGAN (Odena et al. (2016)). DCGAN extends the default GAN, which is a feed forward network, to a deep CNN which enabled much more complex image generation at a higher resolution. ACGAN adds the additional problem of label classification to the end-to-end system by using a labeled dataset to learn conditional image generation. The generator learns to make images based on the input label plus the noise vector while the discriminator for ACGAN learns image classificaion and image validity in parallel. This concept is taken advantage of in the proposed model for this project.

3 Dataset

The data used for training is the ImageNet 2012 (Deng et al. (2009)) image classification dataset. ImageNet one of the most standardized computer vision datasets for both image classification and object detection. In this particular instance of ImageNet, there are 1,000 classes with roughly 1,300 images per class all with different resolutions, making ImageNet also one of the most diverse image classification datasets. The large diversity in images should allow the generator to be able to work well an any image outside of the training data, assuming that the model can become fit to the data. Since image super sampling is a self-supervised machine learning task, the training procedure is fairly trivial. The training data X is the same as the labeled data Y, with the only exception being that the input image will be down-sampled by a factor of 2x. The method for down-sampling will be bi-cubic interpolation (Ledig et al. (2016)) since that is generally the preferred method for reducing the resolution of an image while mitigation information loss.

4 METHODOLOGY

4.1 BASELINE MODEL

The model created for this project was inspired by SRGAN model in proposed in (Ledig et al. (2016)). The main idea of the SRGAN is to use a long sequence of ResNet blocks (He et al. (2015)) to process the low-resolution input image followed by pixel shuffle to up-sample the processed image.

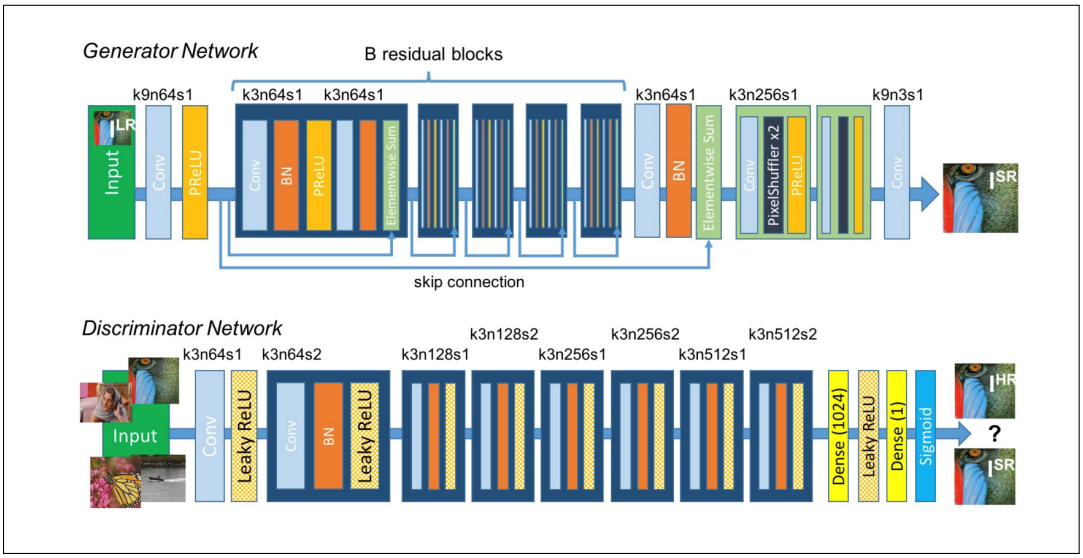

Figure 1: SRGAN architecture of Generator and Discriminator Network with corresponding kernel size (k), number of feature maps (n) and stride (s) indicated for each convolutional layer (Ledig et al. (2016)).

As stated in section 2.1, there are three total loss functions at play when training the generator of the SRGAN model. The loss functions needed for training are formally defined below:

4.1.1 PIXEL-WISE MSE LOSS

$$L_{MSE}(\hat{Y}, Y^{HR}) = \frac{1}{WH}\sum_{i=1}^{W}\sum_{j=1}^{H}(Norm(\hat{Y}_{i,j}) - Norm(Y^{HR}_{i,j}))^2; \quad \hat{Y} = G(\theta_G; Y^{LR}) \tag{1}$$

MSE loss between two images, $\hat{Y}$ and Y^{HR}, is shown above where Y^{HR} is the ground truth HR image and $\hat{Y}$ is the predicted image generated from $G(\theta; X)$ where G is the generator network, θ is the trainable parameters of the function, and X is the LR image. $Norm$ is the normalization function for pixel values of images mapping $[0, 255] \rightarrow [-1, 1]$.

4.1.2 VGG CONTENT LOSS

$$L_{Con}(\hat{Y}, Y^{HR}) = \frac{1}{W_{I,J}H_{I,J}}\sum_{i=1}^{W_{I,J}}\sum_{j=1}^{H_{I,J}}(\phi_{I,J}(\theta_\phi; \hat{Y})_{i,j} - \phi_{I,J}(\theta_\phi; Y^{HR})_{i,j})^2; \quad \hat{Y} = G(\theta_G; Y^{LR}) \tag{2}$$

VGG content loss between two images, $\hat{Y}$ and Y^{HR}, is shown above which looks remarkably similar to the MSE loss. The main difference is that the input images, $\hat{Y}$ and Y^{HR}, are passed through the pre-trained VGG19 model where ϕ is the VGG19 convolutional backbone and θ_ϕ is the respective trainable parameters, making $\phi(\theta_\phi; \hat{Y})$ the feature vector corresponding to the predicted HR image and $\phi(\theta_\phi; Y^{HR})$ is the feature vector corresponding to the ground truth HR image.

4.1.3 ADVERSARIAL LOSS

$$L_{Adv}(Y^{LR}) = \sum_{n=1}^{N} -\log(D(\theta_D; \hat{Y})_{Validity}); \quad \hat{Y} = G(\theta_G; Y^{LR}) \tag{3}$$

The adversarial loss used for updating the generator weights to encourage more realistic looking images. The validity output of the discriminator uses the sigmoid function for probabilistic class mapping, $\sigma(x) = \frac{1}{1+e^{-x}}$, so binary cross entropy i.e., log loss is used to compute the adversarial error. D represents the discriminator network where θ_D corresponds to the trainable variables of the discriminator. It is important to note that the parameters of the discriminator are not updated with this loss function, they are only used to compute the gradients for updating the generator.

4.2 PROPOSED MODEL

The model proposed in this paper is built upon the same network architecture but has a few additions to improve the quality of the generated image. The main contribution to the original model is the U-Net (Ronneberger et al. (2015)) at the base of the generator with the addition of attention modules and pixel shuffle for up-sampling. The main idea behind using a U-Net to preprocess the image is to learn low-resolution embeddings for the input image that can be useful in creating pixel information when the image in further up-sampled. The original model design only allows the model to see the local area around any given pixel in the input image, leading to poor contextual information flow when generating the output image. We can calculate how much information from neighboring pixels is used to compute the pixel value for the final output image; Assuming only 2x up-scaling, 1 (9x9) kernel + 34 (3x3) kernels in low resolution space + 1 (9x9) kernel in high resolution space equates to any pixel in the final image having the ability to gain information about pixels in a 80-pixel radius around it. From a human point of view, it is often beneficial to have high quality information of all sections of an image to be able to understand certain parts of an image. Imagine looking at a heavily zoomed in or cropped part of an image without having any knowledge of the scenery or the general setting of the image. It could be difficult to predict what the cropped section would look like if it were a higher resolution. This is where spatial information becomes useful for image understanding. The hypothesis with the proposed model is that the low-resolution embeddings along with the attention modules in the U-Net will help the model understand the full context of the input image before learning local high-resolution pixel values. In other words, it should be easier for the model to create realistic looking information in the output high-resolution prediction. Another small change to the proposed SRUGAN model is replacing the *B* residual blocks with *B* dense blocks with a higher feature size followed by another *B* dense blocks at the default feature size. DenseNet (Huang et al. (2016)) modules are an extension of the ResNet modules by simply taking advantage of more residual connections while also having a slightly higher feature space. DenseNet modules are able to capture higher dimensional features while also having better gradient updates for deeper networks due to the increased residual connections. The final addition to the SR-UGAN is the inclusion of label predictions shown in ACGAN (Odena et al. (2016)). The labels from ImageNet are used additionally to train the discriminator in parallel with the image validity predictions. The generator is not trained directly on these labels, but is able to accept richer gradient updates from the discriminator which should be learning filters that correlate more in line with the features of what it is classifying. The idea is that these feature-rich gradients will allow the generator to understand what is in the image without having to learn it directly.

The loss functions for training SR-UGAN are defined below:

4.2.1 PERCEPTUAL LOSS

In SRGAN the perceptual loss is the combination of the content loss and the adversarial loss. This is the sum of losses exactly used to train the generator of SRGAN (Ledig et al. (2016)):

$$L_P^{SRGAN} = L_{Con}(\hat{Y}, Y^{HR}) + 10^{-3} L_{Adv}(Y^{LR}) \tag{4}$$

For SR-UGAN, the model performed better with the addition of the pixel-wise MSE loss. This loss also significantly improved training speed at early stages, i.e., generated images at iteration 20 are clearly distinguishable.

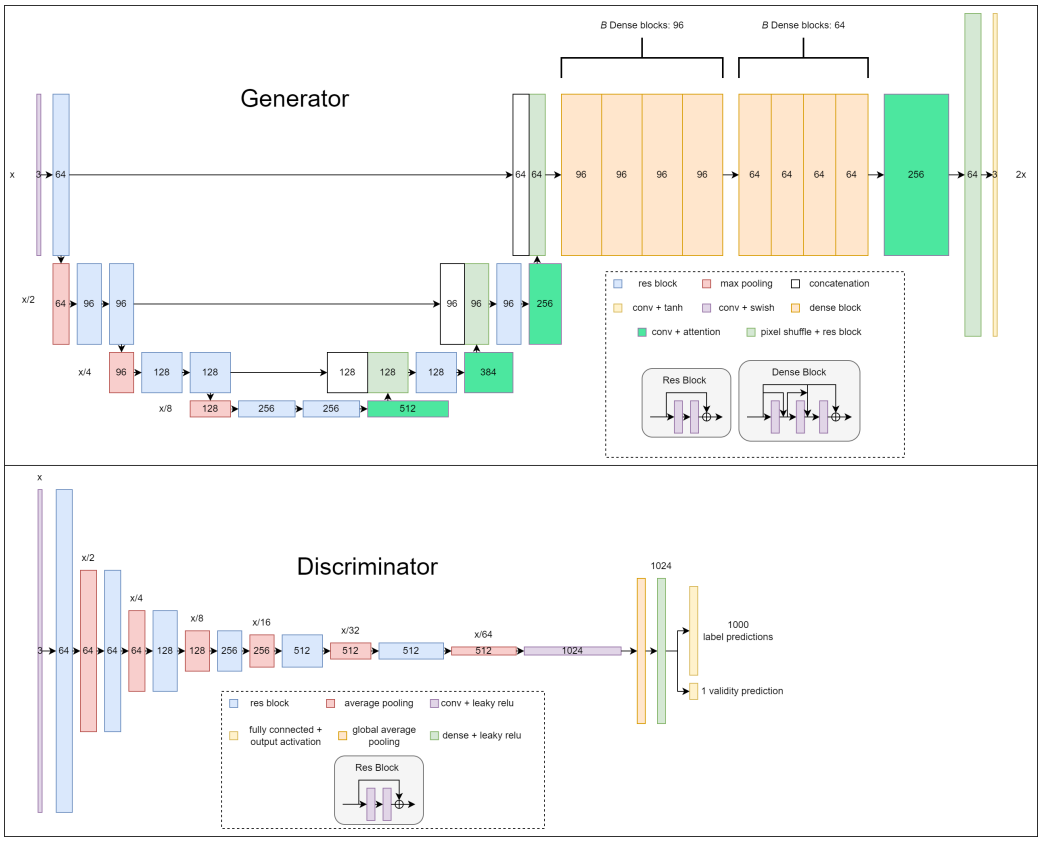

Figure 2: Proposed model SR-UGAN. Generator and discriminator architecture shown where each box in the models contains a number corresponding to the number of channels/kernels/filters in the subsequent layers of the blocks. Attention method used for upsampling is squeeze-excitation and spatial attention. B=8 for the model used in this paper, totalling to 16 dense blocks.

$$L_P^{SRUGAN} = L_{MSE}(\hat{Y}, Y^{HR}) + L_{Con}(\hat{Y}, Y^{HR}) + 10^{-3} L_{Adv}(Y^{LR}) \tag{5}$$

As seen above, the overall loss of the generator networks (perceptual loss) is a combination of the loss functions described in (Ledig et al. (2016)).

4.2.2 VALIDITY LOSS

$$L_V(Y^I, Y^V) = -\sum_{n=1}^{N} Y^V(\log(D(\theta_D; Y^I)_{Validity})) \tag{6}$$

The validity loss is similar to the adversarial loss of the generator. The difference is that the validity loss is used to update the parameters of the discriminator for learning if an input image is real or fake. Y^I is simply any input image where Y^V is the validity label for the image where $Y^V \in \{0, 1\}$.

4.2.3 CLASS LOSS

$$L_{Class}(Y^I, Y^C) = -\sum_{n=1}^{N} Y^C(\log(D(\theta_D; Y^I)_{Class})) \tag{7}$$

The class loss is the categorical cross entropy between the class output of the discriminator given by the softmax function for multi-class probability mapping, $\text{SoftMax}(x_i) = \frac{e^{x_i}}{\sum^C e^{x_i}}$, and the class label vector which is vector of zeros except for the positive class label which is a 1.

4.2.4 DISCRIMINATOR LOSS

In SRGAN the discriminator loss is simply just the validity loss since that is the only model output of the discriminator in SRGAN (Ledig et al. (2016)). SR-UGAN learns to minimize both the validity loss and the class loss in parallel since they are independant of each other:

$$L_D^{SRUGAN} = [L_V(Y^I, Y^V), L_{Class}(Y^I, Y^C)] \tag{8}$$

4.3 TRAINING AND TESTING PROCEDURE

For training, n images from each class in the training dataset were used such that the amount of computer memory used was maximized. In other words, my personal computer did not have enough RAM to fit the entire dataset. 64 GB of RAM was maximized with $n \approx 180$ giving a training dataset of 180,000 images, all HR downsampled to 256x256 pixels and LR downsampled to 128x128 pixels. The initial plan was to leave all images at their default resolution and to generate batches of ragged tensors for training since both generator and discriminator models are build to accept images or arbitrary resolutions. The problem again is a hardware constraint of GPU memory. The machine used for training has a RTX 3080 with 10 GB of RAM, leading to OOM errors if a single image larger than 500^2 pixels was used to compute gradients for the massive generator and discriminator models. In the end, a batch size of 3 was the largest size possible for training, given that the large number of trainable parameters for both models living in GPU memory. For testing, the validation set of ImageNet was used. Since we only care about the performance of the generator during testing, we don't need any labels for the images and we can keep the images at their default resolutions for HR.

To evaluate a super sampling method on a LR image and its expected value HR, the Fréchet inception distance (FID) between predicted images $\hat{Y}$ and the ground truth images Y^{HR} will be used for evaluating the performance of the model predictions against standard image interpolation methods. FID is a metric used to assess the quality of images created by a generative model that compares the distribution of generated images with the distribution of a set of ground truth (HR) images (Heusel et al. (2017)).

$$d_F(\mu, \nu) := \big(\inf_{\gamma \in \Gamma(\mu,\nu)} \int_{\mathbb{R}^n \times \mathbb{R}^n} \|x - y\|^2 d\gamma(x, y) \big)^{1/2} \tag{9}$$

where $\Gamma(\mu, \nu)$ is the set of all measures on $\mathbb{R}^n \times \mathbb{R}^n$ with marginals μ and ν on the first and second factors respectively Heusel et al. (2017).

5 RESULTS AND ANALYSIS

For training, the Adam optimization algorithm (Kingma & Ba (2014)) was used for computing gradients and updating the parameters of both the generator and discriminator. A learning rate of 10^{-4} was used for 100,000 iteration followed by a learning rate of 10^{-5} for another 100,000 iterations. Convergence was never reached in either network, this implies that longer training is needed for these model to reach maximum performance.

	Nearest-Neighbor	Bi-Linear	Bi-Cubic	SR-UGAN
FID	67.618	71.617	63.162	22.123

Figure 3: Table of FID scores correlated to method used to super-sample. All methods other than the SR-UGAN are standard interpolation methods.

The table above shows the FID computed on between up-sampled images and the ground truth HR images. The first three method of up-sampling which are the current standard for image re-scaling are nearest-neighbor, bi-linear, and bi-cubic interpolation, which received a FID of 67.618, 71.617, and 63.162, respectively. The FID between predicted images and HR images is 22.123 which shows that the model prediction is mathematically more similar to the HR images than the interpolated images. The FID between the HR images and themself is zero for reference.

6 CONCLUSION

Deep convolutional generative models have proven to be a superior method up image super resolution as apposed to the traditional methods of image interpolation. The FID for the image distribution generated by the SR-UGAN model is far lower than that of the distributions generated from standard image interpolation algorithms.

7 FUTURE WORK

This study was incomplete mostly due to the hardware constraints mentioned in section 4.3. Image quality of the generator network most certainly would be higher if training with a higher batch size (ideally > 32) as well as increased number of update iterations. Generator performance would also likely increase if trained on images of a higher resolution. It would have also been more scientific to compare the FID results of other deep learning methods such as SRGAN on the validation data set to do a direct comparison of the proposed model compared to previous iteration of deep learning generative models. Development of the SR-UGAN took longer than expected due to long training times. A variation of SR-UGAN that has native 4x up-scaling needs to be tested.

REFERENCES

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

Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(2): 295–307, 2016. doi: 10.1109/TPAMI.2015.2439281.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014. URL https://arxiv.org/abs/1406.2661.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL http://arxiv.org/abs/1512.03385.

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. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/8a1d694707eb0fefe65871369074926d-Paper.pdf.

Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. Densely connected convolutional networks. *CoRR*, abs/1608.06993, 2016. URL http://arxiv.org/abs/1608.06993.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014. URL https://arxiv.org/abs/1412.6980.

Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew P. Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. *CoRR*, abs/1609.04802, 2016. URL http://arxiv.org/abs/1609.04802.

Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans, 2016. URL https://arxiv.org/abs/1610.09585.

Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks, 2015. URL https://arxiv.org/abs/1511.08458.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks, 2015. URL https://arxiv.org/abs/1511.06434.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015. URL http://arxiv.org/abs/1505.04597.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

Zhaowen Wang, Ding Liu, Jianchao Yang, Wei Han, and Thomas S. Huang. Deeply improved sparse coding for image super-resolution. *CoRR*, abs/1507.08905, 2015. URL http://arxiv.org/abs/1507.08905.

8 Supplementary Material

Figure 4: Sample of 3 random images taken from the ImageNet validation set. The images were downsampled by a factor of 2x where the left column shows the bicubic interpolation method, the middle column shows the model prediction, and the right column shows the ground truth HR image.

Figure 5: Sample of 3 random images taken from the ImageNet validation set. The images were downsampled by a factor of 4x where the left column shows the bicubic interpolation method, the middle column shows the model prediction, and the right column shows the ground truth HR image.