Class-Conditional Superresolution with GANs

Vincent Chen *
Stanford University
vschen@stanford.edu

Liezl Puzon *
Stanford University
puzon@stanford.edu

Christina Wadsworth *
Stanford University
cwads@stanford.edu

1. Abstract

We investigated the problem of image super-resolution, a classic and highly-applicable task in computer vision. Recently, super-resolution has been very successful at low upscaling factors (2-4x) with GANs. In this paper, we proposed several methods to introduce auxiliary, conditional information into a super-resolution model to produce images more tuned to the human eye. We showed that our model, trained on the MNIST and CelebA datasets and conditioned on digits/facial attributes respectively, helped constrain the solution-space of the super-resolution task to produce more accurate upscaled images.

2. Introduction

Super-resolution is a highly challenging task that has a wide range of applications, such as facial recognition, satellite imaging, and medical image processing [16]. Super-resolution has been an attractive research topic for the last two decades and has been very successful at low upscaling factors (2-4x) with Generative Adversarial Networks (GANs). The input to a super-resolution GAN is a low resolution image (e.g. 16x16). The output from the GAN is a higher resolution image (e.g. 64x64). We aim to add a class conditional feature to GANs to fine tune results at upscaling factors that GANs are currently fairly successful on.

General information, such as class, may not have a huge effect on many super resolution models because specific, pixel level information and edge information has been found to be more important than what the image is of. Therefore, we focused first on two datasets, MNIST and CelebA, where we believed there was potential for classification to impact the super resolution. Both datasets are discussed more in depth in Section 5.

We believe that our work could have important real-world applications. Imagine a scenario in which a law enforcement agency needs to enhance photographs of suspects captured on low-resolution cameras. By providing information about what we know about the suspect's gender and whether or not they wear glasses, we can produce a more accurate image. The super-resolution task is unique in that we often know information about the images that we want to improve. Our rationale is to take advantage of this assumption and devise ways to introduce known information to improve the quality of super-resolution models.

*indicates equal contribution

3. Related Work

3.1. Non-Deep Learning Approaches

One of the first super-resolution models was Example-Based Super-Resolution [15] that used a nearest neighbors search to sharpen edges and image details in one pass after cubic spline [14] interpolation. Example-Based Super-Resolution was fairly successful at sharpening image edges.

Another model that serves as a baseline today is Lukin et. al's Bicubic Interpolation SR model [8], developed in 2006. Lukin et. al. saw promising results on a scaling factor of 2x, and improved over the previously state-of-the-art bilinear interpolation model [9]. Another non-deep learning approach, [4] utilizes sparse-coding methods that pay attention to learning and optimizing dictionaries/mapping functions of low-resolution patches and aggregating corresponding high-resolution patches for reconstruction.

Generally, non-deep learning approaches have been limited in super-resolution quality, upscaling factor, and generalization across different types of images.

3.2. SRCNN

One of the first promising deep learning super-resolution models was SRCNN [6]. SRCNN [6] used three layers: patch extraction and representation, non-linear mapping, and a final reconstruction layer. SRCNN [6] saw fairly good results up until a 2x upscaling factor, and inspired a number of other CNN based approaches. Other SRCNN inspired models include Pixel Recursive Super Resolution (PRSR) [11] and Perceptual Loss [13]. PRSR saw promising results with an upscaling factor of 4x from 8x8 to 32x32, and Perceptual Loss saw similar results to SRCNN [6], but with three orders of magnitude faster training. Another CNN based approach was a deeper CNN-based model coined VDSR [7]. VDSR cascaded a pair of convolution and non-

linear layers repeatedly, with depth ranging from 5 to 20. As depth increased, so did performance, and VDSR [7] saw a marked improvement over SRCNN [6], especially at the higher (up to 4x) scaling factors.

3.3. Super-Resolution GAN (SRGAN)

In the past few years, interesting progress has been made using GANs [2] towards super-resolution. Ledig [5] proposed a feed-forward network as the generating function that used a perceptual loss that was the weighted combination of several components. In essence, this paper features a deep residual network with the capability for large upscaling factors ($4\times$) with photo-realistic reconstructions of low-resolution images. SRGAN [5] is generally regarded as the state of the art for super-resolution today.

Radford and Metz [10] added a set of constraints to [5] to improve the stability of training GANs to create DCGAN [10]. [10] also saw interesting results with arithmetic properties on faces. For example, [10] showed that smiling woman - neutral woman + neutral man produced the image of a smiling man. [10] performed this arithmetic on mean vectors, but we believe a similar production of specific qualities, such as smiling in this case, could be conveyed with a classifier during the learning process. Discriminative Generative Networks have also seen success specifically on faces datasets. Yu et al. [12] was able to produce 8x super-resolution on faces, albeit somewhat fuzzy. Right now, GANs are widely regarded as producing state-of-the-art super-resolution output.

There are also existing works specifically targeting MNIST and CelebA, the two datasets we focused on. On the MNIST dataset, we adapted CGANs[3], which introduced a conditional component to GANs and was fairly successful generating images in MNIST, to super-resolution factors of 2x and 4x on the MNIST dataset. Additionally, we adapted a SRGAN model [1] for the CelebA dataset by introducing class information in the super-resolution task. Before our changes, [1] saw successful outputs with a 4x scaling factor, as seen in Figure 3.

4. Method

4.1. General Adversarial Networks (GANs)

In 2014, Goodfellow et al.[2] introduced the notion of Generative Adversarial Networks (GANs) which offered a new level of impact for generative models. Under the GAN framework, a generative model competes with a discriminative adversary. The discriminative model determines whether a sample is generated or a data example, while the generative model attempts to fool it. These adversarial models are trained simultaneously.

In some ways, this model can be interpreted as a two-player minimax game:

$$\begin{aligned}\min_G \max_D (D, G) = E_{x\ p_{data}}[\log(D(x)] \\ + E_{z\ p_z(z)}[\log(1 - D(G(z)))]\end{aligned}$$

[2]

4.2. Conditional GANs

The development of Conditional-GANs by Mirza (2014) suggested that we could take advantage of natural one-to-many mappings to produce a conditional predictive distribution for the discriminator *D*/generator *G* network described above [3]. This framework consists of both *D*/*G* models conditioned on some auxiliary information $\mathbf{y}$. This is represented by the following objective function:

$$\begin{aligned}\min_G \max_D (D, G) = E_{x\ p_{data}}[\log(D(x|\mathbf{y})] \\ + E_{z\ p_z(z)}[\log(1 - D(G(z|\mathbf{y})))]\end{aligned}$$

[3]

Mirza & Osindero were able to successful generate digits of the MNIST dataset using the Conditional-GAN framework.

We propose a GAN framework 4.2 for single-image super-resolution, drawing inspiration from the [5] referenced in 3. We use original, high-resolution images (I^{HR}) as true labels and low-resolution images (I^{LR}), downsampled by some factor k from originals, as training examples. With this framework, we explored two separate approaches for introducing class information to the model: using a conditional distribution in the generator and discriminator models 4.3 and adding a class loss term to the generator loss 4.4, as depicted in Figure 1.

4.3. Conditional GAN (CGAN)

We used specific image attributes on our labeled datasets (e.g. classification categories, facial features) as conditional variables, $\mathbf{c}$, for each example or test-time input. As a result, our generator and discriminators were jointly conditioned as $D_{\Theta_D}(x, \mathbf{c})$ and $G_{\Theta_G}(z, \mathbf{c})$, respectively. Recall that in conventional GANs, z is the latent representation [2], initialized to random noise. In our model, however, we used the low resolution image, I^{LR}, as the input to the generator. The input x represents an image, either real or fake (produced by the generator).

The overall objective of our adversarial model is shown below:

$$\begin{aligned}\min_{\Theta_G} \max_{\Theta_D} (D, G) = \\ E_{I^{HR} \sim p_{\text{train}}(I^{HR})}[\log(D_{\Theta_D}(I^{HR}, \mathbf{c})] \\ + E_{I^{LR} \sim p_G(I^{LR})}[\log(1 - D_{\Theta_G}(G_{\Theta_G}(I^{LR}, \mathbf{c})), \mathbf{c})]\end{aligned}$$

[3]

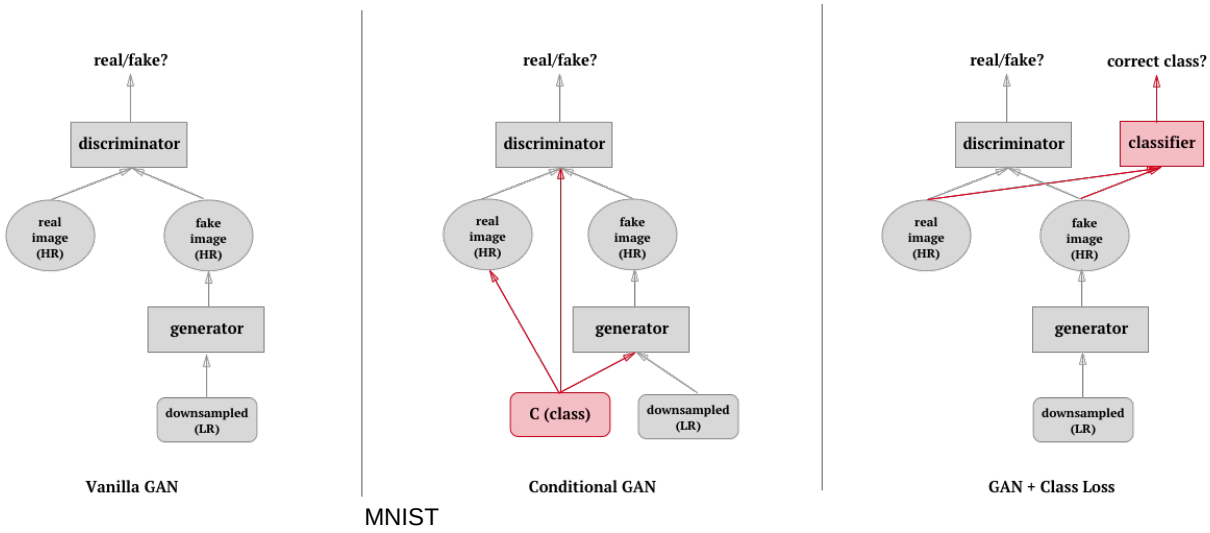

Figure 1. Approach for introducing auxiliary information, indicated in red, compared to Vanilla GAN.

The simplest way to produce the conditional distributions for the generator and discriminator is by feeding the additional class information, c into both networks. We do this by concatenating the c to the inputs in the dense layers of each model.

4.4. GAN + Class Loss

Another way we introduced class information to a GAN was by adding and minimizing an explicit class loss term, as seen in Figure 1. Given that we have an additional computation for the class loss term, $C(z)$, our adversarial model is represented below:

$$\min_G \max_D (D, G) = E_{x\ p_{data}}[\log(D(x)] + E_{z\ p_z(z)}[\log(1 - D(G(z) + C(z)))]$$

To do this, we first trained a classifier on the dataset, based on a set of pre-determined attributes (i.e. gender, eyeglasses). Next, we froze the classifier's weights and imported the frozen graph into the super-resolution GAN model. This was important so that the model did not update the weights for the classifier's parameters during training of the super-resolution model. We then replaced nodes in the classifier to connect the existing GAN graph with the classifier graph in order to use the predicted classification scores in the GAN model.

With the connected classifier model, we computed the scores for the original 64x64 images, I^{HR}. We then computed the scores for the generator's resulting 64x64 images. The original classifier's scores were treated as labels and the generator's scores as the logit values for a softmax cross entropy loss. This resulting class loss term was then added to the generator loss and minimized with an AdamOptimizer in the same way that a vanilla GAN might operate.

By adding this class loss term to the generator loss, the generator optimized for correct prediction on the classifier, which demonstrated results closer to the original 64x64 image. The class loss backpropogated to the generator's parameter weights, such that auxiliary information the classifier presented (e.g. what specifically eyeglasses look like) was learned by the generator. The class loss term served as an indication to the model that the information we classify on was important to learn.

5. Dataset and Features

5.1. MNIST

MNIST is a dataset of hand written numbers 1-9. On this data set, we expect a better performance than baseline super-resolution models due to the highly-recognizable, distinct shape of each digit. In addition, the dataset has a fairly uniform distribution of digits, which will help with learning each condition well. MNIST [17] contains 60,000 training images, some of which can be used for validation purposes, and 10,000 test images, drawn from the same distribution. All are grayscale images of digits, sized normally and centered in a 28x28 size image. We preprocessed the dataset by downscaling by a factor of 4 to use as inputs to our super-resolution models.

5.2. CelebA

Large-scale CelebFaces Attributes (CelebA) [18] contains 202,599 celebrity images, each with 40 attributes, such as "Male" or "Eyeglasses". We split these images up into random training, test, and validation splits ourselves. CelebA offers a pre-processed Aligned and Cropped version of the dataset, which provides 178x218 images that have been aligned and cropped around the upper torso. We

Figure 2. Aligned and Cropped image on the left, preprocessed and cropped 64x64 input on the right.

Figure 3. Results from SRGAN model. Trained by David Garcia. [1]

further pre-processed these images to randomize saturation and contrast. Additionally, we took a random crop around the face, which produced a 144x144 image that we reshaped to 128x128 and downsized to 64x64. We used these 64x64 images as the "true output", or labels, for our models, and used a further downsized 16x16 image (4x downscale) as input.

We are adapting an existing GAN implementation [1] to add attribute conditions. Our approach seeks to improve the recognizability of specific attributes to the human eye, while maintaining the resolution quality of the SRGAN model.

As you can see in Figure 3, the current GAN model outputs a high-resolution image, but the person's face doesn't look like the original. In fact, it seems to portray less feminine qualities than the original image. By adding gender class information, we hope to make these subtle features much more realistic in the generated image.

6. Experiments

6.1. MNIST + CGAN

For our first model, we implemented a vanilla super-resolution GAN by feeding downscaled MNIST images as input (instead of uniformly sampled noise), and adding a content loss term. We then implemented a conditional GAN (CGAN) by modifying the model and feeding in the MNIST class labels as described in Section 4.3. The results are shown in Figure 6.

To qualify the results, we turn to the output examples in Figure 4. As we can see, CGAN produces significantly clearer images than the vanilla GAN. Our results are more unambiguously recognizable as digits. Meanwhile, the vanilla GAN model produced artifacts that create confusion about to the exact identity (or classification) of an digit image, as seen in red in Figure 6.

To quantify the quality of output produced by the MNIST super-resolution Conditional GAN (CGAN), we trained a separate deep, convolutional MNIST classifier on the training data from MNIST. The resulting test accuracy on the MNIST test batch was **0.9925**. We also classified the super-resolved output of the MNIST super-resolution vanilla GAN (using the downscaled test batch images as input), which gave us an accuracy of **0.6953**. We then classified the super-resolved output of the MNIST super-resolution CGAN (using the downscaled test batch images as input), which gave us an accuracy of **0.8438**. In Figure 5, we see that the two images in the top two rows generated by the vanilla GAN were nearly unrecognizable to the MNIST classifier. The first row image generated by the vanilla GAN was misclassified as a 2 with probability **100%**, whereas the true class was 3. The CGAN output was correctly classified as a 3 with probability **58%**. These probabilities indicated that the classifier had a harder time distinguishing the 3 in the CGAN output as opposed to the 3 in the original high-res image (which was classified as a 3 with probability **100%**). However, the CGAN output was much more recognizable the true digit than the vanilla GAN output. The CGAN arguably preserved the overall structure of the original input image when its output is compared to the true high-res image. Similarly, the second row image generated by the vanilla GAN was misclassified as a 7 with probability **94%** whereas the true class is 9. For the bottom two rows, we notice that qualitatively, the CGAN output was much smoother but the classifier was still able to correctly classify both the vanilla GAN output and the CGAN output with 100% probability.

From the confusion matrix7, we can see that the vanilla GAN output was not as recognizable by the classifier since many of points representing the true positive cases (on the diagonal) are lighter than the points representing true positives from the classified CGAN output, especially for 4,5,6, 7, and 9. Interestingly, CGAN-super-resolved 9's are much more often mistaken for 4's than the vanilla GAN output; however, vanilla GAN-super-resolved 9's are much more often mistaken for 7's than the CGAN output. Furthermore, super-resolved 8's are the least recognized digits out of both of the models.

This was a successful quantitative and qualitative confirmation of our hypothesis that introducing class information could help control super-resolution outputs to be more recognizable to the human eye.

6.2. CelebA + CGAN

We implemented a similar conditional GAN architecture for the CelebA dataset as the MNIST dataset conditional GAN as described in 4.2. In this case, concatenating the

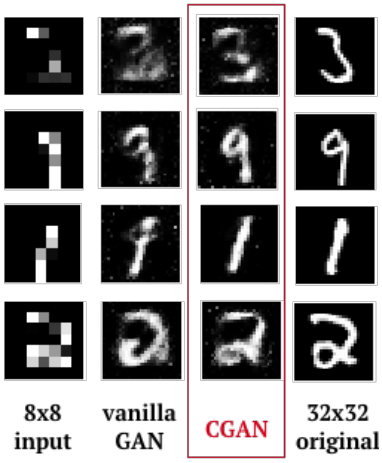

Figure 4. Comparison of vanilla GAN vs. conditional GAN results on MNIST dataset.

Image	True Class	True Class Probability (Original)	True Class Probability (CGAN)	True Class Probability (Vanilla GAN)
1	3	100%	58%	0% (predicted 2 with probability 100%)
2	9	100%	79%	0% (predicted 7 with probability 94%)
3	1	100%	100%	100%
4	2	100%	100%	100%

Figure 5. MNIST classifier results on the original high-res image, the super-resolution CGAN output, and the super-resolution vanilla GAN output.

attributes labels proved more difficult than it was for the MNIST model, for which we implemented a much simpler architecture with fewer convolutional layers.

Figure 9 demonstrates that the original super-resolution GAN trained reasonably well, while the CGAN training (loss plot in Figure 8) was much more unstable. Although we used the same loss function in both the original super-resolution GAN and the CGAN, the CGAN loss moves much more erratically, with the discriminator overpowering the training by becoming very quickly too powerful. This is likely because we had to add a new fully-connected layer to both the generator and discriminator in the CGAN so that we could incorporate the concatenated attribute in the generator and discriminator. The original GAN we used for CelebA super-resolution was a ResNet architecture with no fully-connected layers in the generator and discriminator. This additional complexity likely caused training to be much more sensitive to hyperparameter tuning.

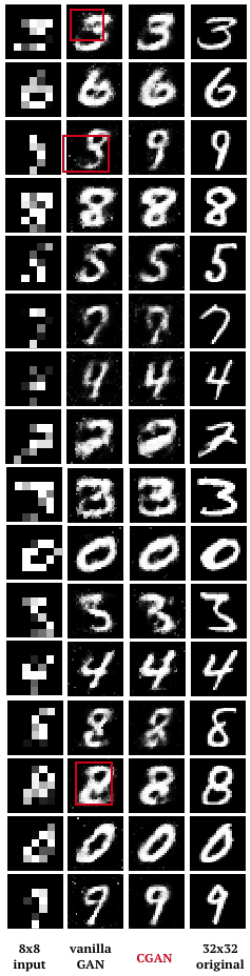

Figure 6. A partial batch of MNIST trials. Artifacts from vanilla GAN highlighted in red.

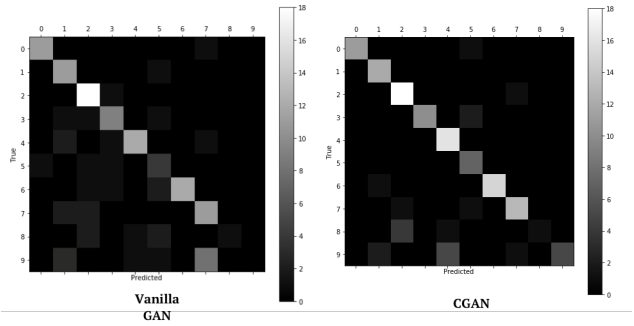

Figure 7. Confusion matrix for the classified vanilla GAN MNIST output (left) and CGAN MNIST output (right).

As a result of these challenges, we were forced to shift our architecture to append the class label to each model input, which led to significant instability during training.

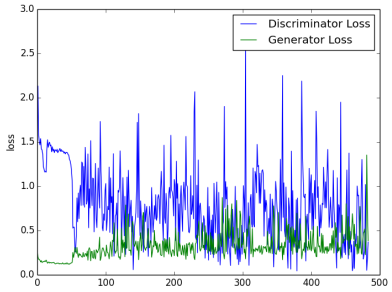

Figure 8. CGAN loss plot for the first 500 batches of training

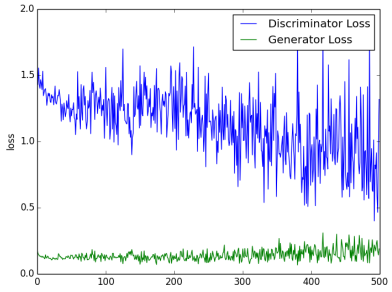

Figure 9. Vanilla GAN loss plot for the first 500 batches of training

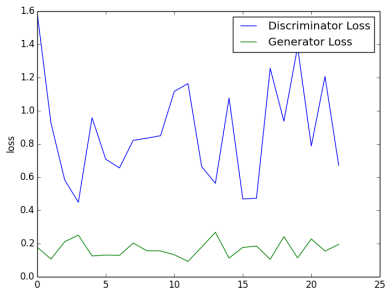

Figure 10. Vanilla GAN Output loss graph

6.3. CelebA + Gender Classifier

First, we trained a Convolutional Neural Network gender classifier on CelebA that we adapted from Github to classify on CelebA [19]. We used a validation split of **.2** and reserved **3000** random images from CelebA for our test set. Our classifier performed at **.938** accuracy on the training set and **.856** accuracy on the test set after 10 epochs. We found that the crop was too small, such the image felt very close to the face and the classifier was unable to use signs such as neck or hair to distinguish between genders. The classifier performed fairly well considering it was essentially only considering facial features and face shape. Our gender

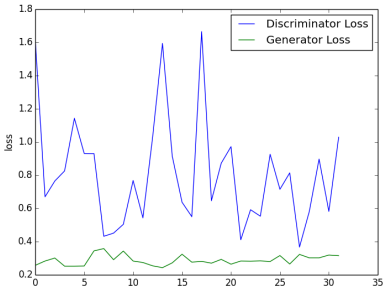

Figure 11. SRCGAN with gender classifier loss graph.

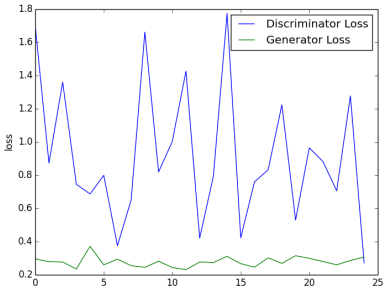

Figure 12. SRCGAN with eyeglasses classifier loss graph.

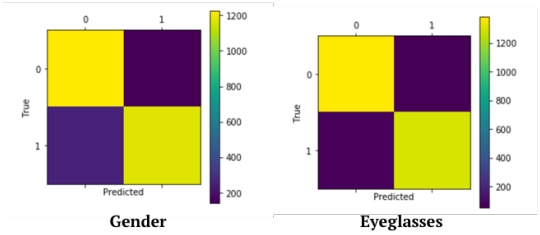

Figure 13. Gender/eyeglasses classifier's confusion matrix on our test set of 3000.

classifier's confusion matrix is depicted in Figure 13.

Next, we froze the gender classifier's weights and imported the graph into the super-resolution GAN. As described in Section 6.2, we used the classifier to calculate a class loss, then we added this term to the generator loss with a `class_loss_weight`. We experimented with different versions of a `class_loss_weight`, and settled on .5 because the results looked best qualitatively.

Overall, the gender class loss term didn't seem to have huge differences in the final outputs. We looked at numerous random samples of 16 images at over 100 images total, and we saw very similar results. Figure 14 depicts a random sample of 5 images.

As you can see, there are few qualitative differences be-

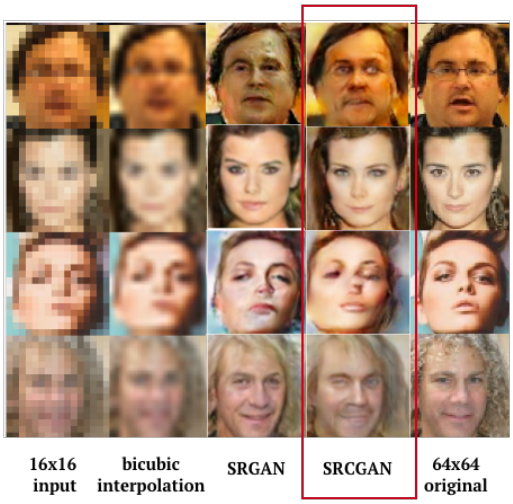

Figure 14. From left to right: 16x16 input image, bicubic interpolation, SRCGAN output, original SRGAN output, original images.

Image	True class probability (SRCGAN)	True class probability (SRGAN)
1	97%	69%
2	93%	98%
3	58%	8%
4	59%	24%

Figure 15. The probabilities of each example image's true class. Our SRCGAN versus the original SRGAN model. Image numbers correspond to the images in Figure 14, from top to bottom.

tween our outputs in column 4 (boxed in red) versus the original GAN's outputs in column 3. Our images appear smoother, but it's hard to tell if there are any true gender differences.

In Figure 15, we do see a few quantitative differences. We have shown the probability of the true class on each of our outputs versus each of the original model's outputs. As you can see, our model's outputs classify better for the bottom two images in particular. For example, we see the man that we have outputted does seem to have a stronger jawline and more of a beard than the original GAN's output, which is reflected quantitatively by the classifier.

We also looked at the specific example that inspired us to start with gender in the first place. Figure 16 depicts the woman that we saw the original model produce the wrong output for. As you can see, our model definitely produces a more feminine face.

Overall, we saw a few minor changes between our model and the original model. However, we didn't see such obvi-

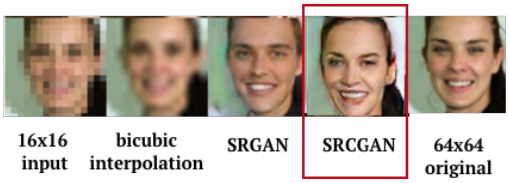

Figure 16. From left to right: 16x16, bicubic interpolation, our output, true output. The woman depicted is the same as in Figure 3.

ous changes that we could conclude that the classifier made a significant difference. We think this is because gender is such a broad classification that the model would have trouble learning a specific thing to improve on. Super-resolution models generally look for specific, local information to produce a higher resolution image, and our hypothesis is that something like gender is simply too broad to learn effectively, across the board. In order to test this hypothesis, we turned to something more specific: eyeglasses.

6.4. CelebA + Eyeglasses classifier

We used the same classification CNN model, except we swapped the gender labels for eyeglass labels. Our classifier produced a train accuracy of **1.0** and a test accuracy of **0.956**. The classifier's confusion matrix and example wrong classifications can be found in Figure 13.

After running the GAN with the same hyperparameters that we used on the CGAN with gender, we saw promising results. As you can see in Figure 18, the glasses we generated seem clearer and covered the entire eye, especially in the first and third examples. In the second example, we see a stronger glasses frame as well. Since eyeglasses is such an obvious characteristic, our quantitative results were not helpful, as each image classified as **0.99** probability of eyeglasses, as we would probably expect. However, we can see qualitatively from the results that our output has a stronger emphasis on fully-formed glasses. Figure 18 depicts three random samples, but this was consistent across the random batches we looked at.

Overall, we saw marginally better glasses across the board. We hypothesize that this was because eyeglasses are a very specific and easily-perceivable attribute, and therefore the model is more easily able to learn how to successfully form a more simple characteristic. Already though, the previous model is fairly good, so we only saw minor improvements.

7. Conclusion

Our MNIST experiments were very successful. We saw outputs consistent with the original, high-resolutionimages that were unambiguously recognizable as digits. With the conditional information, our model was able to learn how

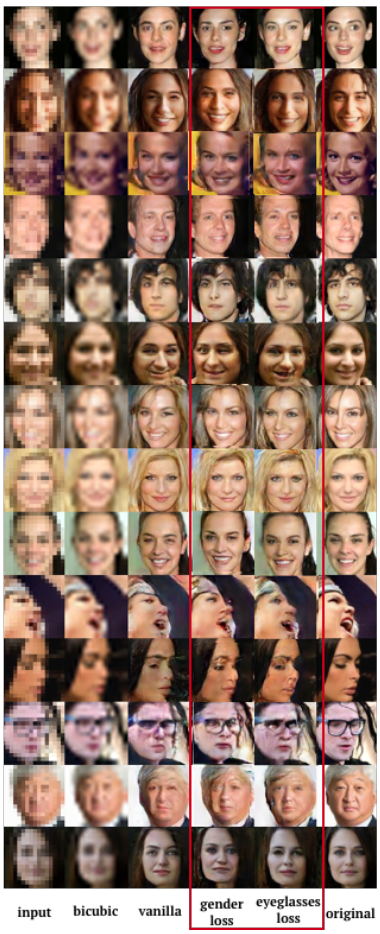

Figure 17. Full, random batch showing both gender and eyeglasses classifiers compared to vanilla model.

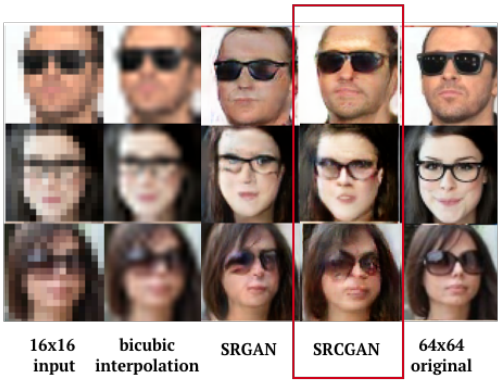

Figure 18. Three images with eyeglasses chosen at random. From left to right: 16x16 input, bicubic interpolation, our SRCGAN output, SRGAN output, true 64x64 image.

to represent more clearly formed digits.

During our CelebA experiments, we saw that the additional information provided helped the SRGAN make minor changes towards better output. With additional gender information, the changes were more vague, so it was harder for us to conclude that our model was having a large effect, although we did see improvements in some cases. With additional glasses information, the model more obviously learned how to form better shaped glasses and made minor tweaks across the board.

Overall, class conditioning seems promising in cases where there are clear cut attributes that the model is not learning towards as much as it could be (e.g. a fully formed number or wearing eyeglasses). On specific datasets, SRCGAN seems to be a promising super-resolution model. With more time, we would hope to better develop a general conditional GAN that can be used for super-resolution on a variety of tasks.

Moving forward, it would be interesting to attempt conditional super-resolution with multiple attributes concurrently. That is, indicate that a person is wearing glasses *and* male. In addition, it may be interesting to try different models for introducing auxiliary information (i.e. InfoGAN [20]) to see which model performs best with the super-resolution task.

References

[1] David Garcia. *Suepr Resolution on Celebrity Faces Github*. https://github.com/david-gpu/srez

[2] Goodfellow, I., et. al (2014). Generative Adversarial Nets.

[3] Mirza, M. & Osindero, S (2014). Conditional Generative Adversarial Nets.

[4] Wright et al. (2010) Image Super-Resolution via Sparse Representation. *IEEE transactions on image processing*, 1-13.

[5] Letdig et. al (2016). Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network.

[6] Dong, Chao, et al. "Learning a deep convolutional network for image super-resolution." European Conference on Computer Vision. Springer International Publishing, 2014.

[7] Kim, Jiwon, Jung Kwon Lee, and Kyoung Mu Lee. "Accurate image super-resolution using very deep convolutional networks." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.

[8] Lukin, Alexey, Andrey S. Krylov, and Andrey Nasonov. "Image interpolation by super-resolution." Proceedings of GraphiCon. Vol. 2006. 2006.

[9] Yuan, Shuai, et al. "High accuracy wadi image interpolation with local gradient features." Intelligent Signal Processing and Communication Systems, 2005. ISPACS 2005. Proceedings of 2005 International Symposium on. IEEE, 2005.

[10] Radford, Alec, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks." arXiv preprint arXiv:1511.06434 (2015).

[11] Dahl, Ryan, Mohammad Norouzi, and Jonathon Shlens. "Pixel recursive super resolution." arXiv preprint arXiv:1702.00783 (2017).

[12] Yu, Xin, and Fatih Porikli. "Ultra-resolving face images by discriminative generative networks." European Conference on Computer Vision. Springer International Publishing, 2016.

[13] Johnson, Justin, Alexandre Alahi, and Li Fei-Fei. "Perceptual losses for real-time style transfer and super-resolution." European Conference on Computer Vision. Springer International Publishing, 2016.

[14] R. Keys, Cubic Convolution Interpolation for Digital Image Processing, IEEE Trans. Acoustics, Speech, Signal Processing, Vol. 29 No. 6, 1981, pp. 11531160.

[15] Freeman, William T., Thouis R. Jones, and Egon C. Pasztor. "Example-based super-resolution." IEEE Computer graphics and Applications 22.2 (2002): 56-65.

[16] Nasrollahi, Kamal, and Thomas B. Moeslund. "Super-resolution: a comprehensive survey." Machine vision and applications 25.6 (2014): 1423-1468.

[17] Deng, Li. "The MNIST database of handwritten digit images for machine learning research [best of the web]." IEEE Signal Processing Magazine 29.6 (2012): 141-142.

[18] Ziwei Liu and Ping Luo and Xiaogang Wang and Xiaoou Tang. "Deep Learning Face Attributes in the Wild." Proceedings of International Conference on Computer Vision (ICCV). December, 2015.

[19] Robert Coleman, Tensorflow Image Classification. https://github.com/rdcolema/tensorflow-image-classification

[20] Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., & Abbeel, P. (2016). InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets.

[21] JJ Allaire, Dirk Eddelbuettel, Nick Golding, and Yuan Tang (2016). tensorflow: R Interface to TensorFlow. https://github.com/rstudio/tensorflow

[22] Code from CS 231N A3 was used