BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Improving the Speed of Image Super-Resolution Diffusion Models

Dominik Klement, Jozef Makiš, Peter Zdravecký

1. Introduction

Image super-resolution is a technique for generating high-resolution images consistent with their low-resolution pairs. Numerous techniques have been used in the past, such as: Bicubic scaling, iterative methods, Variational Auto Encoders (VAE), Conditional Generative Adversarial Networks (CGANS). VAEs and GANs produce fairly good results, but they cannot replicate tiny details, or are hard to train (i.e. mode collapse problem with GANs). The authors of [1] proposed a solution based on a Diffusion Model (DM) conditioned on an original low-resolution image. They achieved incredible level of details both on faces and wild images (imagenet), which proved that DMs can be used for super-resolution task and outperform previous sota methods.

However, DMs are slow, because a denoising model, usually U-Net containing millions of parameters, is executed hundreds to thousands times during the inference for each input image. To speed up the inference, we decreased the U-Net size. We further combined it with features of the conditional image from a pretrained VGG-11 model. This way, U-Net only needs to learn how to use those features, which is usually easier than learning the features from scratch.

Our experiments showed, that we were able to reduce the U-Net size by 7x and the inference time by 1.77x with reasonable quality loss. Also, our proposed method of adding VGG features to the U-Net improved the quality of the samples produced by the small denoising model and increased its convergence rate.

2. Diffusion Speed Improvement

A diffusion model is trained to model the distribution of the input data by generating a sample from a Gaussian noise using iterative denoising process. Usually, U-Net is used to predict the amount of noise in the input sample, which is then subtracted from the input sample to obtain a denoised input sample. The authors of [1] put a bicubically scaled low-resolution image as an input and let the DM iteratively refine the image details by conditioning (i.e. concatenating the conditional with the noisy image along channel dimension) the U-net with the low-resolution image.

However, DMs require running multiple (potentionally thousands) of iterations of U-Net containing millions of parameters, which makes them slow in comparison with other methods. To reduce the inference time of the proposed method, we decided to dramatically decrease the denoising model size. To counteract the reduced capacity, we added the inner features of the conditional image from a pretrained VGG-11 model to the features of the denoising mode. The conditional image does not change during the denoising process, which means that we can evaluate the VGG-11 only once during the whole inference and cache the features. This way, U-Net does not need to learn complex features of the conditioning image, which increases the model convergence and the output quality.

2.1 U-Net and VGG

The authors of the original method used ResNet blocks inside the unet and applied attention on the bottleneck layer (schematics can be seen in Figure 1). They used 5 blocks with channel multipliers: 1,2,4,8,8, where the base number of channels was 64. We, on the other hand, got rid of the last two blocks, which resulted in a 7x smaller U-Net.

We propose two modifications of the denoising U-Net:

- Addition - we added VGG features to the outputs of the resnet blocks as shown in Figure 1
- Concatenation - we concatenated VGG features with the outputs of the resnet blocks along the channel dimension and applied 1x1 conv layer to match the number of channels in the U-Net.

The first methods is a more natural way of inserting features, as it does not require any additional transformations of the feature space because the VGG and U-Net tensor dimensions match exactly. On the other hand, concatenation of the features increases (in our case doubles) the number of output channels. One possible way to deal with this problem is to increase the number of channels in all subsequent layers, which would result in larger U-Net network. We dealt with this problem by applying 1x1 convolutions without any additional nonlinearities to preserve the channel dimensions.

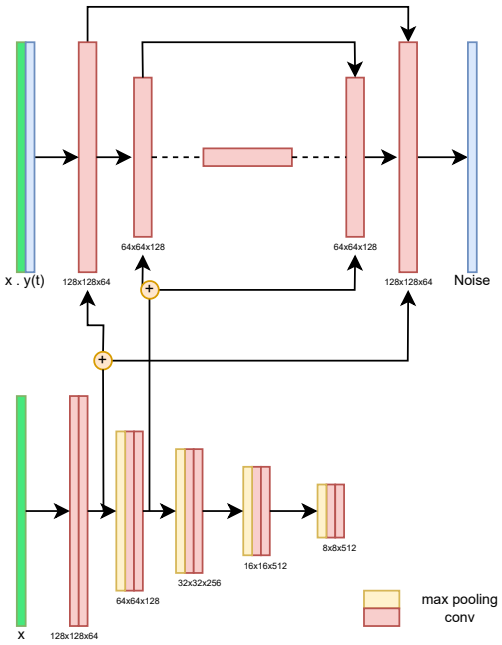

Figure 1. UNet schematics.

3. Experiments

For our experiments, we decided to use a public available Github[1] repository containing the code for the DM and U-Net. We then modified and combined it with VGG model.

We trained the diffusion models on a single Nvidia A100 GPU for 500K iterations with batch size 128 from scratch. Other training parameters were not changed. As the loss function, we used L1 loss as in the original paper. We used FFHQ dataset and split it to 55K training images and 10K evaluation images. The low-resolution image is 32x32 pixels and we upscaled it 4x; hence, the upscaled image is 128x128 pixels.

Furthermore, we took a VGG-11 model pre-trained on the Imagenet dataset. We also experimented with other VGG model versions, but the results were almost the same, so we sticked with the smallest one.

3.1 U-Nets

We wanted to explore multiple U-Net sizes. We firstly chose the original one, containing 66M parameters. As a second model, we halved the number of parameters by getting rid of the last resnet block and the attention layer. As the smallest model, we got rid of the last two resnet blocks of the original U-Net, which resulted in only 9M of parameters. We tried two versions of the small 9M parameter model, with and without attention applied to the bottleneck layer, but the attention mechanism did not improve sampling quality.

Furthermore we experimented with added and concatenated features from VGG-11. We concatenated the features only to the small model, as we wanted to improve its sampling quality and kept others as a slightly modified references. We also tried larger VGG models, like VGG-19, but we did not see any improvements over the VGG-11 model, so we continued only with the smallest VGG version.

Lastly, we decided to completely omit the concatenated conditional image from the U-Net and let it only use the VGG features. This also served as a proof that the U-Net does not simply discard the information provided by the VGG model and is able to take advantage of it.

3.2 Evaluation

For evaluation and comparison of the proposed approaches, we decided to use human fool-rate evaluation. We displayed a low-resolution image in the center and two upscaled images on the left and on the right. One of them was the original one and the other was generated. We let people look at the images for 5 seconds and they had to pick the real image afterwards.

To further compare the models, we use visual inspection of the generated images to closely asses strengths and weaknesses of the U-net modifications. Also, to show and compare the model convergence, we used validation PSNR score.

4. Results

To show the outcomes of the models, we focused only on the face photos, as they contain a lot of semantic details that can show whether the model is able to understand the underlying features or not.

[1] https://github.com/Janspiry/Image-Super-Resolution-via-Iterative-Refinement

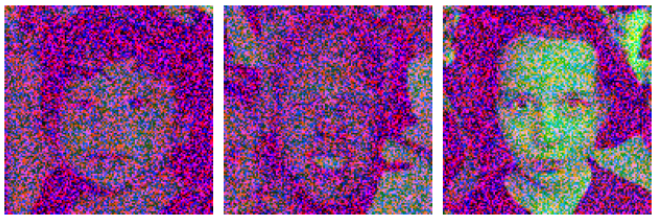

Figure 2. Results of unsuccessful concatenation of features from VGG11.

4.1 Unsuccessful Feature Concatenation

One of our attempts was to concatenate features and then apply 1x1 convolutions to change the number of channels as described previously. During training, we observer that the network was unable to use the information from the VGG itself and failed to learn at all, even if the conditional image was a part of the U-Net input. Figure 4 show the results after 135K iterations during which other proposed models already produced reasonable results. It seems like it discarded a large portion of the information hidden in the conditional image and mainly worked with the input noise only.

We think the failure is either due to the 1x1 convolutions that failed to combine the features from U-Net and VGG, or that a mistake was made during the model development. We would like to examine this wrong behaviour further in the future.

Therefore, in the following results, we only present the second approach of adding VGG features to U-Net along the channel dimension, which we described previously.

4.2 Convergence

Firstly, as can be seen in Figure 3, the original and the half size model converged to the best PSNR scores, although other models are not much behind. The small model with added VGG-11 features converged faster, which proves that the VGG features give the model more information from the beginning and it is easier for the model to build upon those features.

4.3 Face Images

Furthermore, Figure 4 shows the original ground truth images, and figures 5, 6, 7 show how the three main models are able to generate various photos of people faces. Those images were generated using 2000 denoising iterations and it can be seen that all of those models produce authentic images with minor differences that can be spotted after looking at the images for a while - e.g. the woman on the topmost second picture from the left was generated with shadow on her forehead being slightly off by the small model,

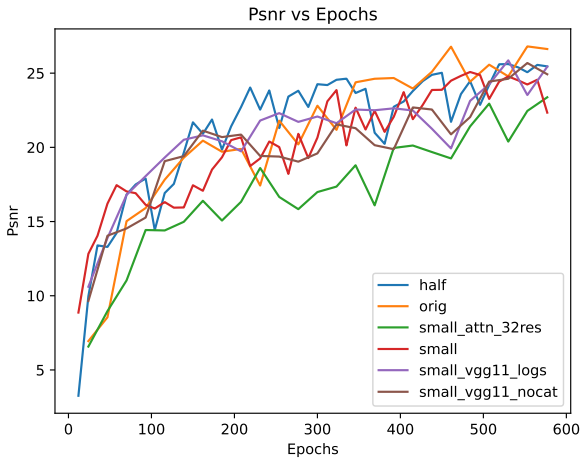

Figure 3. Validation PSNR comparison between all models.

Figure 4. Ground truth high resolution images.

whereas the original and the small model with VGG features generated this correctly. On the other hand, all three models failed to generate the nose ring in the nose of the woman on the center picture, which is likely caused by the distribution of the training data (i.e. not many people modifies their face that way). Overall, it can be seen that the small model with VGG features generates samples closer to the original model, which proves that the proposed method enhances the U-Net.

4.4 Different Number of Iterations

Furthermore, Figure 10 shows comparison of different numbers of inference interations. It can be seen that the original U-Net model performs the best among all three models. On the other hand, the small model deviates from the other two in the first 500 itera-

Figure 5. Images from original model.

Figure 6. Images from small model.

tions, but after 1000 and 2000 denoising iterations, the results are close. It can also be seen that the small model is less consistent with the woman's eyes, but the one with VGG features corrects this mistake. Moreover, the small model with VGG features outperforms the original one in terms of the quality of woman's eyes. The left eye (from our perspective) is much more blurry and the iris is not rendered well by the original model.

However, the small model with VGG features produces worse mouth pictures. It can be seen on the woman's face in Figure 10 that her lips are larger than they should be after 500 iterations, then the mistake is

Figure 7. Images from smaller model with VGG11.

fixed after 1000 iterations and reappears again after 2000 iterations, which means that the model is not stable in generating authentic mouth.

4.5 Fool Rate

To test the systems in real world, we let 10 participants score photos as described earlier. It supports our findings that the original U-Net 66M params model achievs the best score and beats all other models. As Figure 8 shows, we were able to achieve fool rate of 40%, which is different from the sota score the authors of [1] achieved, but we were upscaling images from 32x32 to 128x128, whereas they upscaled 64x64 image to 256x256, which means that the input image contains more details the model can learn and generate the details from.

Also, the participants were fooled 4% more by the small model with VGG features than by the small model alone, which achieved fool rate of 32% only.

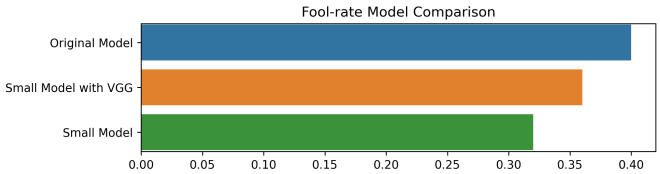

Figure 8. Fool rate comparison.

4.6 Performance

Finally, Figure 9 shows the speed comparison between the original 66M parameters U-Net model and the small 9M parameters model. Despite the original model being 7x larger in its size, the small model

performs only 1.77 faster than the original large model. This disproportion is likely caused by the structure of the U-Net itself, some memory bottlenecks or compute bottlenecks.

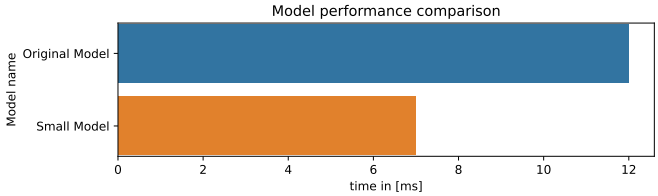

Figure 9. Model speed comparison. The time corresponds to a an avg single forward pass.

5. Conclusion

In this work we first shrank the size of the U-Net model in order to decrease the inference time, which resulted in a U-Net network with decreased capacity. To counteract this effect, we added a pretrained VGG-11 model features of conditional image to the U-Net features. It improved the U-Net performance with no slowdown, because the VGG model needs to be evaluated only once for a single image, because the conditional image does not change throughout the denoising process. Finally, we obtained a model that was 7x smaller and performed 1.77x faster inference than the original model with reasonable loss in performance (sometimes negligible).

As a future work, using these approaches, we can jointly pass the gradients to the VGG model and adjust its features to better suit the U-Net. Also, we can shrink the U-Net model more, identify the performance bottlenecks and try to remove them to speed up the inference process even further.

Acknowledgements

We would like to thank Dr. Hradiš for valuable suggestions and consultations.

Computational resources were provided by the e-INFRA CZ project (ID:90140), supported by the Ministry of Education, Youth and Sports of the Czech Republic.

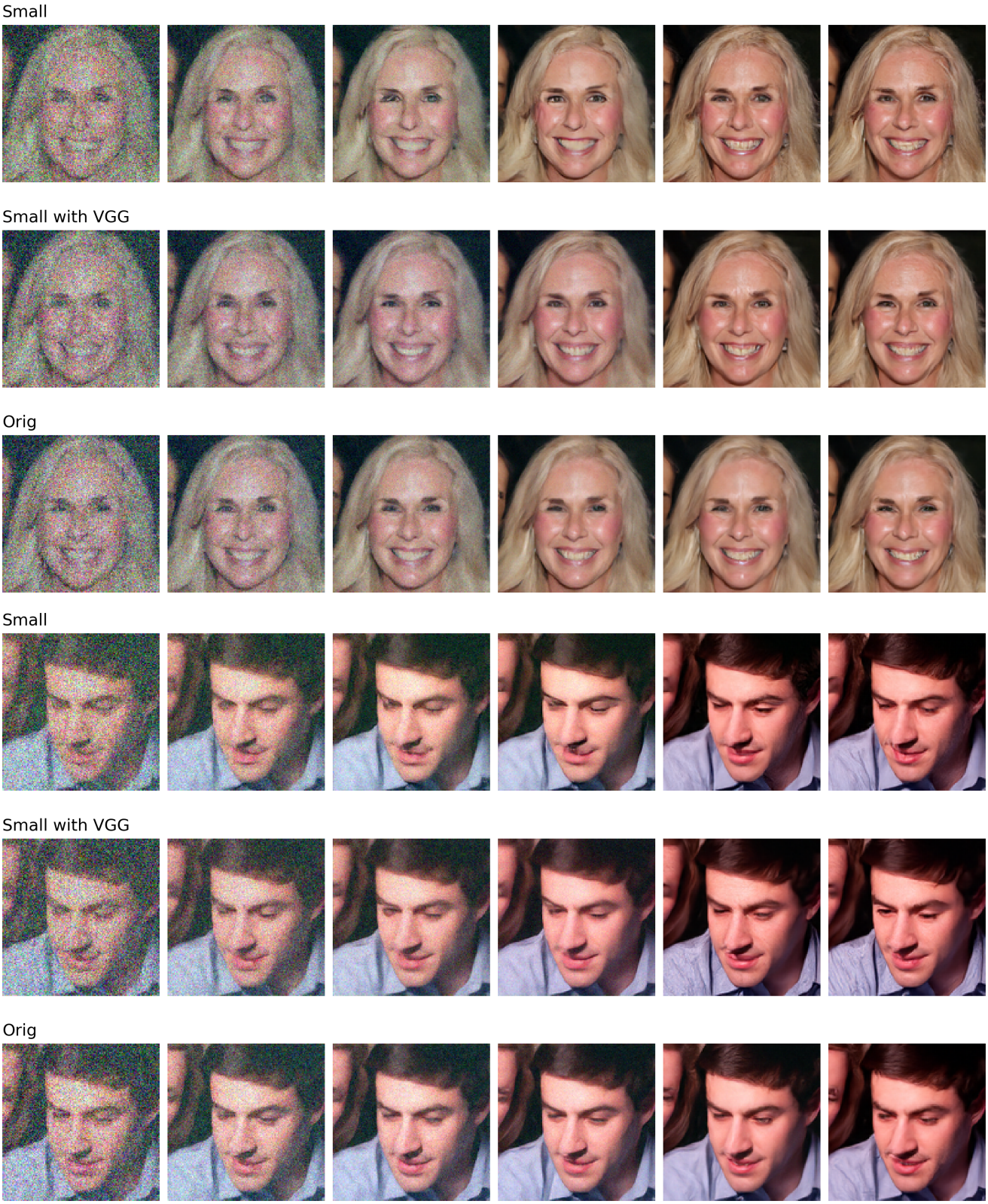

Figure 10. Comparison of different results from small, small with vgg, and original U-Net model. Rows corresponds to different models, and each column corresponds to different number of inference steps: 50, 100, 200, 500, 1000, 2000 respectively.

References

[1] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J. Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4713–4726, 2023.