Edge Guided Progressively Generative Image Outpainting

Han Lin, Maurice Pagnucco, Yang Song
School of Computer Science and Engineering, University of New South Wales, Australia
hantao.lin@student.unsw.edu.au, morri@cse.unsw.edu.au, yang.song1@unsw.edu.au

Abstract

Deep-learning based generative models are proven to be capable for achieving excellent results in numerous image processing tasks with a wide range of applications. One significant improvement of deep-learning approaches compared to traditional approaches is their ability to regenerate semantically coherent images by only relying on an input with limited information. This advantage becomes even more crucial when the input size is only a very minor proportion of the output size. Such image expansion tasks can be more challenging as the missing area may originally contain many semantic features that are critical in judging the quality of an image. In this paper we propose an edge-guided generative network model for producing semantically consistent output from a small image input. Our experiments show the proposed network is able to regenerate high quality images even when some structural features are missing in the input.

1. Introduction

Image repair is an important task within image processing with a wide range of applications including image super-resolution, image inpainting, noise reduction and image expansion. Among them, image expansion and image inpainting tasks are the most challenging due to the requirement to regenerate missing content from a restricted input. While traditional patch-based methods [2, 8] and diffusion based methods [4] have been proposed to solve this problem, the quality of the output is often questionable due to the heavy reliance on existing information in the input image. So these methods would not be able to repair features that are not present in the input. Hence traditional methods are generally more suitable for background repairing, as the entire background is usually a repeating pattern of a smaller region. On the contrary, human face image reconstruction is more difficult, which usually requires the generation of features that do not exist in the input image while keeping the output semantically accurate.

To tackle the weaknesses of traditional methods, deep learning approaches are recently becoming the dominant approach for image repairing. The Generative Adversarial Network (GAN) [6] was firstly proposed to generate images from random one-dimensional input. Recent developments of GAN variants have been widely applied to different areas including style-transfer [33], image synthesis [3], text-to-image generation [21] and image inpainting [31, 18]. Other techniques such as WGAN [1] have also been proposed to improve the stability of training to prevent model collapse.

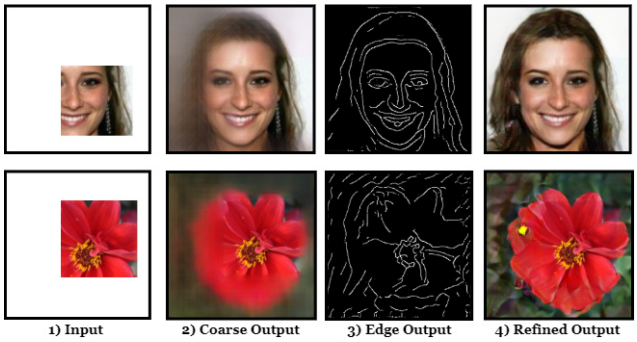

Figure 1. Illustration of the outputs from our proposed network. Images from left to right are: (1) Input images with missing content. (2) Coarse outputs from the first stage network. (3) Edge maps from the second stage network generated based on the coarse output. (4) Final outputs from the third stage network.

Inspired by the well-known autoencoder network structure, recent image inpainting networks often adopt an encoder-decoder convolutional network structure [31] to improve performance. In contrast, a simple Convolutional Neural Network (CNN) [14] would often result in blurry outputs which are much more inferior to the ground truth. To improve this, as neighbouring pixels around the missing area typically sit in the middle of the input and prediction, attention based techniques [11] are often used to improve the semantic coherence of the output. In addition, while image-inpainting tasks often assume the missing regions are much smaller than the input information, more recent neural network structures [24] have been proposed attempting to regenerate images under more challenging conditions where the input is a minor proportion of the missing region,

as illustrated in Figure 1. This problem is often referred to as *image outpainting* or *image extrapolation*. For such image outpainting tasks, the network has to predict and refill the missing region with features totally absent from the input. This raises two main challenges for the outpainting network:

- The network needs to know which features are missing and how they should be located in the output relative to the spatial location of other features.

- The conditional input can be spatially distant from the missing regions to be predicted. It is difficult to make predictions for them due to the lack of neighbouring ground truths.

To solve the above issues, we propose a three-stage image outpainting neural network inspired by the recent work of EdgeConnect [18], Lafin [28] and Semantic Regeneration Network (SRN) [24] to generate semantically coherent images with clear boundaries between the main object and background. The network consists of three stages where the first stage is a coarse generator, the second stage is an edge image generator and the last stage is a refinement generator which combines the output from previous stages to create the final output. Each of the generators follows an encoder-decoder structure with dilation blocks and adversarial learning in the latter two generators. We evaluated our model quantitatively and qualitatively on the CelebA-HQ [13] and Oxford flower102 [19] datasets with 256x256 resolution. Our experimental results show that the proposed approach is able to achieve more stable and visually convincing outputs against other state-of-the-art models.

Our main methodological contributions are:

- We propose a 3-stage deep learning model that contains a dedicated edge-generator to improve the sharpness and boundary of objects in the final output. Different from EdgeConnect that trains the edge generator using ground truth images for image inpainting, we introduce edge information into image outpainting and use our edge generator as the second step by using the coarse result as input.

- Our method achieves improved performance and stability with a lower reconstruction loss. Specifically, this is achieved by calculating losses from a pre-trained VGG network at both coarse and refinement stages, whereas the EdgeConnect and SRN only evaluate reconstruction losses at the last stage. Bringing the reconstruction loss forward to the first stage reduces the blurriness on the coarse output which in turn improves the edge output and the final image outpainting result.

2. Related Work

2.1. Image Inpainting

The current mainstream image inpainting methods can be classified into two categories: traditional algorithm based methods and learning-based methods with generative models. Diffusion based methods [4] take local features and fill the missing region from its neighbouring information. Patch-based algorithms [2] would instead complete the missing part by looking for input regions that are similar to the missing region. Despite the differences between the two methods, they both lack the capability to regenerate structures that are not present in the corrupted input. We often see poor performance of inpainting human faces because it usually fails to regenerate semantic structures that are expected to exist. This property makes traditional methods most suitable for images with repeating patterns such as natural scenes or object textures, but a poor choice for human image regeneration tasks.

Recent learning-based methods often carry an encoder-coder structure to predict high-level features with incomplete input. However, the simple autoencoder structure often causes the output to be blurry. Since GAN-based methods were first introduced in [20], various techniques have been proposed to solve this problem. More than one discriminator can be used in the network as proposed in [9] to improve both the local and global image quality. Partial Convolution [16] was proposed to enhance the output quality with irregular shaped masks. Contextual Attention Network [31] and Gated Convolution Network [30] also uses an attention-based coarse-to-fine network structure with global and local critics to produce less blurry images while preventing semantic distortion around the boundary of the missing region. In addition to static image inpainting, an extension on repairing video sequences [27] also achieves good results using coherent features between two frames.

EdgeConnect [18] predicts a Canny edge map of the entire image as part of the input to the second-stage generator to provide auxiliary guidance for image inpainting. The step of converting input to a Canny-edge image can also been seen as similar to the image-to-image translation networks represented by pix2pix [10] and CycleGAN [33]. However, our experiments show that the edge generator lacks the ability to predict an edge map for large missing blocks, which implies the limitation of applying EdgeConnect to image outpainting tasks.

2.2. Image Extrapolation and Outpainting

Image outpainting or extrapolation is considered a more challenging task than image inpainting as the input is even smaller. For example, with a 256x256 pixel image, only a small region of 64x64 pixels is given while the rest is masked. Therefore, the challenge is that the network has to

predict semantic structures with very limited input. While similar tasks have been explored in earlier studies of photo uncropping [32, 23], traditional methods often fail to regenerate high-level features of human faces despite the models being able to blend images with different view angles and appearances. Learning-based approaches on the other hand would not rely on external image-sets to expand from the input. Earlier work [29] demonstrates the possibility to complete images irrespective of the input mask shape by progressively blending similar images into the input. The Recurrent Feature Reasoning (RFR) network [15] also achieves outstanding results by progressively repairing missing regions through multiple stages of the neural network.

Furthermore, due to the one-sided property of image outpainting (i.e., the prediction only expands outwards from the cropped input), weight masks are often used to indicate lower confidence on pixels further away from the input. In SRN [24], the Relative Spatial Variant (RSV) loss was proposed to describe different levels of confidence by assigning weights based on Gaussian filtering. The second stage refinement network in SRN further refines the output from the first stage network to reduce blurriness and improve texture details. Intuitively, the SRN network is similar to the Contextual Attention Network [31] but with improved stage-one and stage-two networks to cope with the more restricted input. Our proposed model uses an extra edge network between the two stages to provide the refinement stage with extra guidance on the shape and location of features.

3. Our Proposed Network

We propose a three-stage image outpainting network that consists of three neural network modules: a coarse generator, an edge generator and a refinement network. Figure 1 shows examples of the expected input and output from the proposed network. Stage one network adopts an autoencoder structure whereas both stage two and stage three networks are based on the GAN architecture. Hence the entire network architecture would have one autoencoder generator A_1 for the stage-one coarse network, one pair of generator G_2 and discriminator D_2 for stage-two and another pair G_3 and D_3 for stage-three. An overview of the architecture of the network is shown in Figure 2.

The autoencoder in our stage one coarse generator uses two down-sampling layers and four dilation blocks with a factor of two. The generators G_2 and G_3 also use a similar encoder-decoder structure with eight dilation blocks. The technique of using dilation blocks was introduced in [9] to replace the fully connected layer in order to promote a larger receptive field at the output neuron. The two discriminators D_2 and D_3 follow the structure of PatchGAN [10] which treats the image as a Markov Random Field (MRF) and then classifies whether the image is real or fake by averaging results from several smaller patches. Each discriminator has five convolutional layers with a stride of two for the first three layers and stride of one for all the other layers.

3.1. Coarse Generator

The stage one coarse generator is to produce an outpainted output from the corrupted input, aiming to achieve an overall semantically coherent visual appearance in the output without emphasising the structural details. Formally, denote the original RGB image as $\mathcal{I}_{\text{gt}}$ and the two-dimensional mask as M. Then the cropped input is $\tilde{I} = I_{\text{gt}} \circ (1 - M)$ where $\circ$ is the Hadamard product operator. Denoting the coarse output as $\mathcal{O}_{\text{coarse}}$, the goal of the coarse generator A_1 can then be formulated as:

$$O_{\text{coarse}} = \bar{I} \circ M + I_{\text{gt}} \circ (1 - M) \tag{1}$$

where $\bar{I}$ is the predicted image from A_1. For training A_1, we explicitly use reconstruction loss consisting of $\ell 1$ loss, RSV loss and perceptual loss from a pre-trained VGG network on ImageNet.

Our coarse network adopts an autoencoder structure similar to the Feature Expansion Network (FEN) proposed in SRN [24]. Different from FEN which only uses the RSV loss and $\ell 1$ loss for training, we also include the perceptual loss using a pre-trained VGG-19 network to further improve the performance. Adding a discriminator would not be helpful for this stage as there is still distinct difference between the coarse image and ground truth image. The output from this coarse generator will then be processed by the next two stages to create better texture and details.

RSV Loss. The Relative Spatial Variant (RSV) loss was first proposed in [24] to overcome the common issue of lacking information for outpainting tasks. As the input patch is only the minority of the whole ground truth image, pixels that are far away from the input are more difficult to predict than neighbouring pixels of the input. This is different from other image-inpainting tasks where the prediction area usually sits close to the input. To intuitively illustrate this, imagine if half of a person's nose is given as input, it is relatively easy to predict the other half of the nose but will be more difficult to predict the person's hairstyle. In order to cope with such input constraints, a Gaussian filter is applied to assign descending weights to all predicted pixels around the input. The formula for RSV loss is thus defined as:

$$\mathcal{L}_{\text{rsv}} = ||(I_{\text{gt}} - \bar{I}) \circ M_{\text{w}}|| \tag{2}$$

where the confidence weighted mask $\mathcal{M}_{\text{w}}$ is

$$M_{\text{w}} = \frac{(g * \bar{M}_w^{c-1}) \circ M}{max((g * \bar{M}_w^c) \circ M, \epsilon)} \tag{3}$$

where g is the Gaussian filter and:

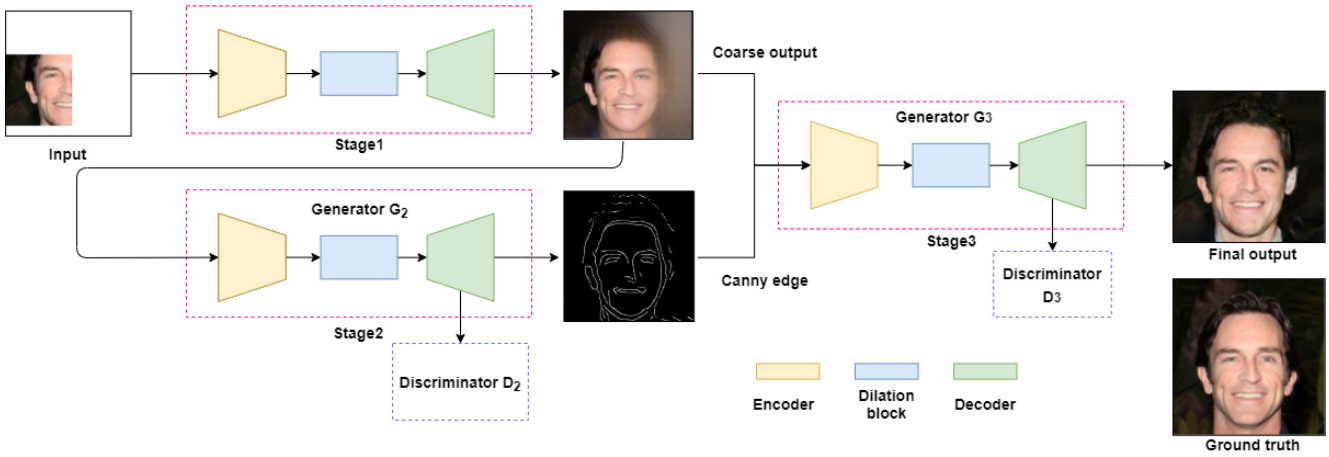

Figure 2. Illustration of the proposed network structure. The entire network contains three modules: (1) Stage one: a coarse generator with an encoder-decoder structure. (2) Stage two: the edge map generator following a GAN structure. It takes the stage-one output as input and returns a predicted edge map. (3) Stage three: the refinement network which also follows a GAN structure. It takes the output from the previous two stages and processes them into a single output image.

$$\bar{M}_w^i = 1 - M + \bar{M}_w^{i-1} \tag{4}$$

The constant parameter c specifies the number of iterations required to calculate the mask, which is set to 9 in our study.

Perceptual Loss. The perceptual loss [12] is computed by computing the difference in VGG-19 activation maps between the ground truth image $\mathcal{I}_{\text{gt}}$ and predicted image $\mathcal{I}$. Our experiments show that adding perceptual loss for training the coarse generator would improve both the quality and stability of the coarse output. Formally, the perceptual loss is defined as:

$$\mathcal{L}_{\text{perc},1} = \mathbf{E}\Big[\sum_i \frac{1}{N_i}||\phi_i(I_{\text{gt}}) - \phi_i(\bar{I})||\Big] \tag{5}$$

where ϕ_i is the activation map of the i^{th} layer and N_i is the number of elements in that layer.

By combining the $\mathcal{L}_1$ loss, RSV loss and perceptual loss, the total loss for the stage-one coarse network is:

$$\mathcal{L} = \lambda_{\ell 1}\mathcal{L}_1 + \lambda_{perc}\mathcal{L}_{perc} + \lambda_{rsv}\mathcal{L}_{rsv} \tag{6}$$

3.2. Edge Generator

The edge generator stage is to generate edge images from the coarse output with a GAN generator. The goal for this stage is to predict the edge map as similar as possible to the ground truth edge map. We would then use this generated edge map for stage three to provide guidance on the shape of generated features and improve image sharpness. The grayscale coarse output O_{gray} converted from O_{coarse} is used as the input for this stage. For training the edge generator, we obtain the ground truth by creating edge maps using standard edge detectors on the ground truth (original complete) images. We have experimented using the Sobel filter and Canny edge detector and we found that the Canny edge detector gives more realistic output, as will be shown in the later section. Nevertheless, regardless the type of edge detector we adopt, the edge generator G_2 is an image-to-image translation network to convert the coarse output to the desired edge map. This process is formulated as:

$$O_{edge} = G_2(O_{gray}, I_{edge}, M) \tag{7}$$

where I_{edge} is the ground truth edge map and M is the same mask as in stage one to merge the predicted edge map with ground truth.

Layer	Layer type	Hyperparameter
1	LayerA	$k=(7,7), s=1, p=0, C_{In}=4, C_{Out}=64$
2	LayerA	$k=(4,4), s=2, p=1, C_{In}=64, C_{Out}=128$
3	LayerA	$k=(4,4), s=2, p=1, C_{In}=128, C_{Out}=256$
4-11	ResNet	$k=(3,3), s=1, d=2, C_{In}=256, C_{Out}=256$
12	LayerB	$k=(4,4), s=2, p=1, C_{In}=256, C_{Out}=128$
13	LayerB	$k=(4,4), s=2, p=1, C_{In}=128, C_{Out}=64$
14	LayerB	$k=(7,7), s=1, p=0, C_{In}=64, C_{Out}=1$

Table 1. Generator architecture.

Layer	Layer type	Hyperparameter
1	LayerC	$k=(4,4), s=2, p=1, C_{In}=2, C_{Out}=64$
2	LayerC	$k=(4,4), s=2, p=1, C_{In}=64, C_{Out}=128$
3	LayerC	$k=(4,4), s=2, p=1, C_{In}=128, C_{Out}=256$
4	LayerC	$k=(4,4), s=1, p=1, C_{In}=256, C_{Out}=512$
5	LayerC	$k=(4,4), s=1, p=1, C_{In}=512, C_{Out}=1$

Table 2. Discriminator architecture.

The generator has 14 layers in total including 3 encoder layers, 3 decoder layers and 8 residual blocks between the encoder and decoder. The encoder has 2 downsampling layers with stride 2 and kernel size 4. Similarly the decoder has

its 2 upsampling layers with same stride and kernel size. Hence the number of filters for each layer of encoder is 64, 128 and 256; and, 256, 128 and 64 for the decoder layers. Spectral normalisation [17], InstanceNorm and ReLU are also applied to the encoder and decoder preventing sudden change in parameter values. Denoting the encoder layers as LayerA and the decoder layers as LayerB which uses transposed convolution, the generator architecture is presented in Table 1. The discriminator layer is similar but uses LeakyRelu to replace InstanceNorm and ReLU. Denoting discriminator layers as LayerC, Table 2 shows the network architecture of the discriminator.

Adversarial Loss and Feature Matching Loss. The total loss for training the GAN model is the weighted sum of the standard adversarial loss $\mathcal{L}_{adv}$ and a feature-matching (FM) loss $\mathcal{L}_{FM}$:

$$\mathcal{L}_{edge} = \lambda_{FM}\mathcal{L}_{FM} + \lambda_{adv}\mathcal{L}_{adv} \tag{8}$$

where λ_{FM} and λ_{adv} are the weight parameters for each of the two losses and

$$\begin{aligned}\mathcal{L}_{adv} = E_{(I_{edge},O_{gray})}\Big[logD_2(I_{edge},O_{gray})\Big] \\ +E_{O_{gray}}log\Big[1-D_2(O_{edge},O_{gray})\Big]\end{aligned} \tag{9}$$

$$\mathcal{L}_{FM} = E\Big[\sum_{i}^{L}\frac{1}{N_i}||D_2^i(I_{edge}) - D_2^i(O_{edge})||\Big] \tag{10}$$

where D_2^i is the activation map for the i^{th} layer of D_2. Here the FM loss shares the same concept as the perceptual loss where it computes the distance between two activation maps of the ground truth and edge output. However, FM loss compares the difference on the discriminators' activation maps between the ground truth and predicted image. A larger FM loss means the features on the predicted image are not so similar to the ground truth.

3.3. Refinement Network

The third stage refinement network generates the final output image by integrating the coarse output and edge map into a single image. Since the coarse output obtained from stage one is often blurry, we incorporate the predicted edge map from stage two to guide the network generating more detailed features, such as textures and shapes of facial structures. By combining the outputs from the previous two stages, the coarse output provides an approximate shape and colour of the image, and the edge map plots out a more precise spatial structure for each feature. We design another GAN model including a discriminator D_3 to determine whether the refined output is real or fake. For this refinement stage, while G_3 and D_3 have similar structures to G_2 and D_2, the generator G_3 requires 4 input channels in order to include the extra dimension from the edge map. Denoting the merged four dimensional input (O_{edge}, O_{coarse}) as O_{merged}, the objective of generator G_3 can be formulated as:

$$O_{fine} = G_3(O_{merged}, M) \tag{11}$$

We use total loss with formula:

$$\mathcal{L}_{fine} = \lambda_{adv}\mathcal{L}_{adv} + \lambda_{perc}\mathcal{L}_{perc} + \lambda_{style}\mathcal{L}_{style} + \lambda_{\ell 1}\mathcal{L}_{\ell 1} \tag{12}$$

which combines the adversarial loss, perceptual loss, style loss and $\mathcal{L}_1$ loss. As the main objective of this stage is to refine the coarse content rather than making prediction from ground up, these losses can help refine the details to make the output more natural and crisp.

Adversarial Loss and Perceptual Loss. We use adversarial loss for training the generator using the critics received from the discriminator, ideally we want our final output to not be distinguishable by discriminator for whether it is real or fake. Perceptual loss on the other hand compares the ground truth against our projected output by comparing their activation map under a pre-trained VGG-19 network. We aim to make the projected output looking similar to the real image by including both losses. The perceptual and adversarial losses for the refinement network are computed similarly to the previous stages, where:

$$\begin{aligned}\mathcal{L}_{adv} = E_{(I_{gt},O_{merged})}\Big[logD_3(I_{gt},O_{merged})\Big] \\ +E_{O_{merged}}log\Big[1-D_3(O_{fine},O_{merged})\Big]\end{aligned} \tag{13}$$

$$\mathcal{L}_{perc,2} = E\Big[\sum_{i}\frac{1}{N_i}||\phi_i(I_{\text{gt}}) - \phi_i(O_{fine})||\Big] \tag{14}$$

Style Loss. The style loss proposed in [22, 5] is the squared Frobenius norm of Gram matrix on the pre-trained VGG network to enhance image quality by reducing the checkerboard artifact in the output. Denoting ϕ as the Gram matrix of activation map, the loss is computed as:

$$\mathcal{L}_{style} = E\Big[||G_2^{\phi}(O_{fine}) - G_2^{\phi}(I_{gt})||\Big] \tag{15}$$

4. Experiments

4.1. Experimental Setup

The proposed model is evaluated on the CelebA-HQ [13] and Oxford Flower102 [19] datasets. For each dataset, we first resize the original images to 256×256 pixels as our ground truth. We then randomly crop the ground truth images to 128×128 pixels to allow the network making prediction based on the cropped area. This means our input is 1/4 of the size of the original ground truth. For each dataset, we use a train/val/test ratio of 0.8/0.1/0.1 with randomly shuffled data.

We train each stage of the proposed model separately and sequentially. We have obtained similar results to [18] where training all 3 stages simultaneously does not improve the performance. Therefore, we prefer to train our networks individually for simplicity.

We use the Adam optimiser for our training with $\beta_1 = 0$ and $\beta_2 = 0.9$. The learning rate is 0.0001 for generators and 0.00001 for discriminators. We compare our results both qualitatively and quantitatively against other state-of-the-art image inpainting and outpainting models including EdgeConnect and SRN.

4.2. Qualitative Results

The comparison of our model against other state-of-the-art models is shown in Figure 3. We can see that our model has better performance than EdgeConnect which is primarily used for small area image inpainting. Our model can also achieve better results than SRN given its finer detail in topological features. There is also less blurriness in the background with a clear boundary between the main object and background. Our proposed model also produces better symmetry for the eyes compared to SRN and hence largely improves the overall quality of images.

Figure 4 shows the comparison between the coarse output and refined output. It can be seen that the coarse stage helps predict the overall shapes and colours of the missing features, whereas the refinement stage focuses on improving the detailed structure and textures.

4.3. Quantitative Results

	CelebA-HQ		
Network	PSNR	SSIM	FID
EdgeConnect[18]	12.88	0.5776	32.43
SRN[24]	15.56	0.6345	34.89
Ours	14.53	0.6022	28.48

Table 3. Comparison between EdgeConnect, SRN and our model on the CelebA-HQ dataset.

4.3.1 Reconstruction-based Evaluation

In our experiment, we use peak signal-to-noise ratio (PSNR), structural similarity index measure (SSIM) [25] and Fréchet inception distance (FID) [7] as our evaluation metrics to be compared against other models. The results are shown in Table 3. While the quantitative results do not show a significant difference among the compared models, our model achieves a better FID score than SRN and EdgeConnect. On the other hand, our model has slightly underperformed SRN on both SSIM and PSNR. However, as mentioned in [31] and [24], these metrics are not ideal for precisely evaluating image inpainting or outpainting tasks as they fail to reflect the actual quality of textures and structural details in the generated images. Therefore, such quantitative results should be used as a reference only to show our model performance when most scores only fluctuates within a small range among different models.

4.3.2 Classification-based Evaluation

For the Oxford Flower dataset, we evaluate the quality of our outpainting results by classification accuracy using the open-sourced VGG-S[1] model. Our intuition is: given a classification model trained on the original Oxford Flower dataset with high accuracy, the classification model should also obtain high accuracy on our outpainted images if the quality of the output is good enough to replicate the original flower type. Therefore, to conduct the experiment: 1) we need to first train the classification model with the original Flower dataset; 2) test the trained model using the original data; and, 3) test the trained model using the outpainted images. Table 4 shows the classification results using both sets of data.

	Original	Regenerated
Accuracy (%)	93.15	82.34

Table 4. Classification accuracy of original and regenerated Oxford Flower data using VGG-S.

We can see from the results that the regenerated data still maintains a relatively good classification accuracy compared to the original data. This means the VGG-S network has good robustness, but more importantly, our outpainting network is able to generate images similar to its original label. This can be attributed to the fact that our outpainting output has a good colour consistency similar to the original image and it is able to replicate the shape and details of the original image in order to make the generated images correctly classified. We also note that this experiment might have a big variation depending on the classification model. As part of the future work, to make this evaluation more persuasive, ideally the experiment should be conducted with several other different classification models to eliminate the model variations.

4.4. Ablation Studies

4.4.1 Edge Detector

In addition to the Canny Edge detector, we have experimented using the Sobel filter for edge map generation. One potential advantage of using Sobel filter over Canny edge is that it allows for different pixel intensities (gradients) to be

[1] https://github.com/jimgoo/caffe-oxford102

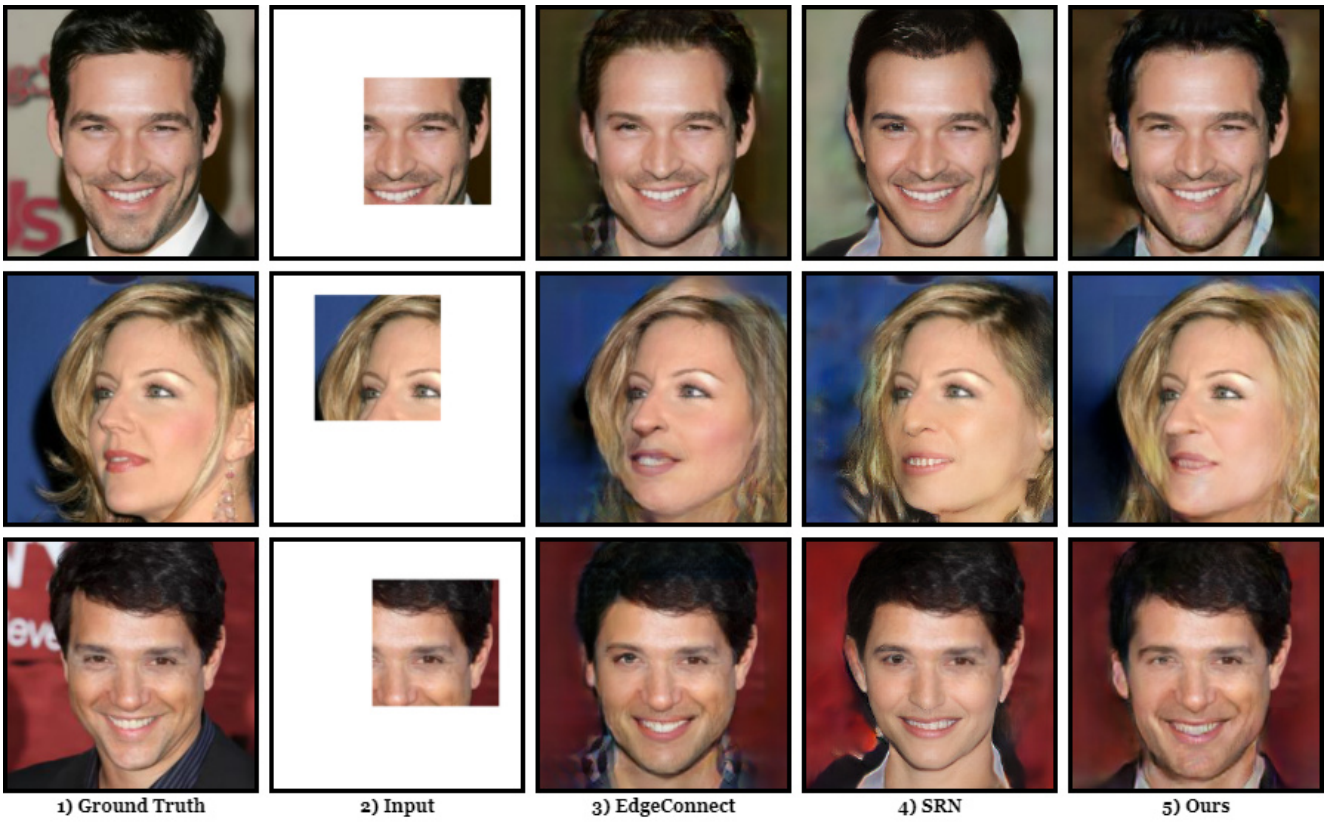

Figure 3. Comparison between different image inpainting and outpainting models. From left to right: 1) ground truth; 2) input; 3) EdgeConnect[18] output; 4) SRN[24] output; 5) our result.

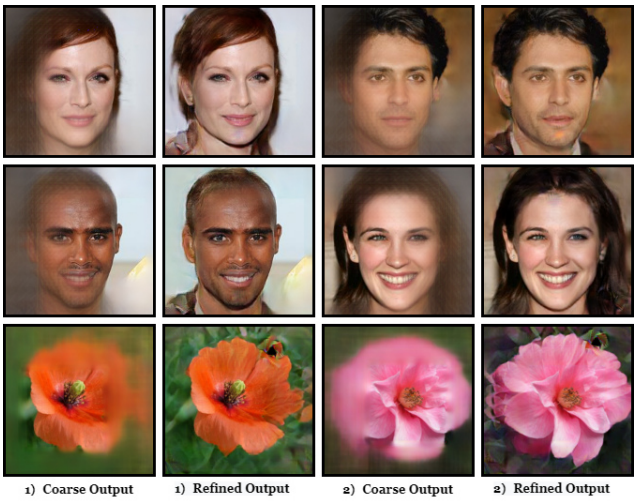

Figure 4. Comparison between coarse output and refined output.

shown on the edge maps. Ideally, such variation should offer extra guidance to the stage three network when predicting missing features. However, our experiment shows the Sobel filter usually fails to provide enough information on the overall structure and shape of the object as we expected.

With Figure 5, we observe that the Sobel edge generator is mainly handling an image-to-image translation task similar to CycleGAN instead of predicting the ground truth edge map. In addition, an image-translation network for Sobel filtering would be more difficult to train as even the ground truth edge can be blurry when the object has low contrast with the background. On the other hand, with Canny edge detector, all edges are sketched with the same intensity and this enforces the network to perform prediction using coarse images.

Based on the quantitative results in Table 5 for final-stage output, Sobel filter actually shows a better the reconstruction-based metrics score than Canny edge because, again, the quantitative metrics are not the best for evaluating image outpainting performance. In addition, the Canny edge detector gives lower FID scores, which is consistent with our qualitative evaluation, and this further demonstrates that FID is a more reliable metric for the image outpainting task.

4.4.2 Loss Function for Coarse Network

The performance of the coarse network output is crucial to the quality of the final output. Different from EdgeConnect

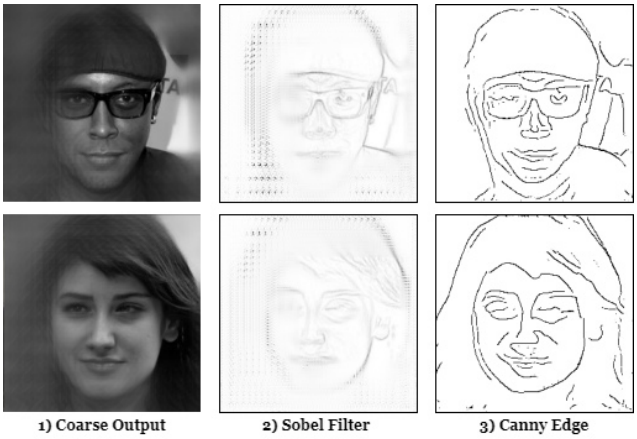

Figure 5. Comparison between the Sobel filter and Canny edge detector. From left to right: the predicted coarse output; output using Sobel filter; output using Canny edge detector. We see the shape of faces are not clearly depicted by the Sobel-based generator.

	CelebA-HQ			Flower102		
Network	PSNR	SSIM	FID	PSNR	SSIM	FID
Sobel	14.59	0.5615	34.73	14.25	0.6412	63.15
Canny	14.53	0.6022	29.48	13.98	0.5731	58.35

Table 5. Quantitative results comparing the Sobel filter and Canny edge detector.

and SRN, we introduced perceptual loss as part of the loss function for the coarse generator. As image outpainting networks are generally difficult to train due to the large missing areas, using the pre-trained VGG network's activation map to compute the loss can help improve the coarse generator. As a comparison, Figure 6 shows samples of coarse output without using perceptual loss. We can see that the coarse image is more blurry with jagged edges as the vanilla $\mathcal{L}_1$ loss does not preserve fine textures very well [10]. Hence by incorporating the pre-trained VGG-19, we add the perceptual loss to help the network learn semantic features and details in order to provide better inputs for the later two stages.

4.4.3 Additional Results and Discussion

One limitation of our model would be its inferior performance on predicting small objects. Figure 7 shows a few results using the CUB200 [26] dataset with our proposed model. From the outputs we can see our model is not well predicting the structure of birds with significant artifacts. Due to the different shapes and sizes of birds even of the same species, the edge generator is unable to delineate the shape of features such as eyes and beaks precisely enough. We would like to improve our coarse and edge generator to better handle small object regeneration. Furthermore, CUB200's small dataset size may also not be sufficient to train our model. Hence how to effectively address small training data with large variation would be a good future direction for our research.

Figure 6. Sample coarse outputs with and without perceptual loss. Top: without perceptual loss; Bottom: with perceptual loss. The performance of the coarse network without perceptual loss is much worse, represented by the jagged edges and inconsistent colours.

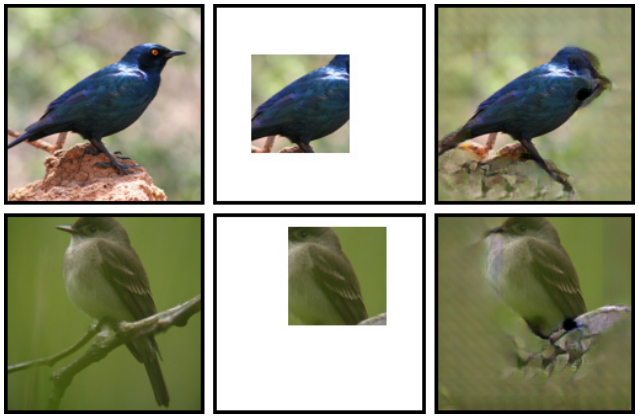

Figure 7. Sample outputs using CUB200, showing the limitation of our model for small object structures.

In addition, the proposed model currently only accepts a square-shaped image patch as input. We will investigate generalising our model to accept irregular shaped input as our future work.

5. Conclusions

We propose a three-stage image-outpainting network to regenerate images from small cropped inputs. The proposed model is able to correctly recreate semantic structures with coherent details by employing a three-stage network structure containing a coarse generator, an edge map generator and a refinement generator. We have evaluated our model against other image inpainting and outpainting models and demonstrated better performance using our method. As part of future work, we will explore new metrics for quantitative evaluation to better reflect the quality of image outpainting.

References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. *ICML*, pages 214–223, 2017. 1

[2] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics*, 28(3), 2009. 1, 2

[3] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *ICLR*, 2018. 1

[4] Alexei A. Efros and William T. Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 341–346, 2001. 1, 2

[5] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *CVPR*, pages 2414–2423, 2016. 5

[6] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014. 1

[7] 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 *Proceedings of the 31st International Conference on Neural Information Processing Systems*, page 6629–6640, 2017. 6

[8] Jia-Bin Huang, Sing Bing Kang, Narendra Ahuja, and Johannes Kopf. Image completion using planar structure guidance. *ACM Transactions on Graphics*, 33(4), 2014. 1

[9] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and Locally Consistent Image Completion. *ACM Transactions on Graphics*, 36(4):1–14, 2017. 2, 3

[10] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, pages 1125–1134, 2017. 2, 3, 8

[11] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and koray kavukcuoglu. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, pages 2017–2025, 2015. 1

[12] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, Lecture Notes in Computer Science, pages 694–711, 2016. 4

[13] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. *ICLR*, 2018. 2, 5

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012. 1

[15] Jingyuan Li, Ning Wang, Lefei Zhang, Bo Du, and Dacheng Tao. Recurrent feature reasoning for image inpainting. In *CVPR*, pages 7760–7768, 2020. 3

[16] Guilin Liu, Fitsum A. Reda, Kevin J. Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. 2

[17] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *ICLR*, 2018. 5

[18] Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Qureshi, and Mehran Ebrahimi. Edgeconnect: Generative image inpainting with adversarial edge learning. In *ICCVW*, pages 1–10, 2019. 1, 2, 6, 7

[19] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Proceedings of the 2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722––729, 2008. 2, 5

[20] Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting. *CVPR*, pages 2536–2544, 2016. 2

[21] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *ICML*, pages 1060–1069, 2016. 1

[22] M. S. M. Sajjadi, B. Schölkopf, and M. Hirsch. EnhanceNet: Single image super-resolution through automated texture synthesis. In *ICCV*, pages 4501–4510, 2017. 5

[23] Qi Shan, Brian Curless, Yasutaka Furukawa, Carlos Hernández, and Steven M. Seitz. Photo uncrop. In *ECCV*, volume 8694 of *Lecture Notes in Computer Science*, pages 16–31, 2014. 3

[24] Yi Wang, Xin Tao, Xiaoyong Shen, and Jiaya Jia. Wide-context semantic image extrapolation. In *CVPR*, pages 1399–1408, 2019. 1, 2, 3, 6, 7

[25] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 6

[26] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-ucsd birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010. 8

[27] Rui Xu, Xiaoxiao Li, Bolei Zhou, and Chen Change Loy. Deep flow-guided video inpainting. *CVPR*, pages 3723–3732, 2019. 2

[28] Yang Yang, Xiaojie Guo, Jiayi Ma, Lin Ma, and Haibin Ling. LaFIn: generative landmark guided face inpainting. *arXiv preprint*, 2019. 2

[29] Raymond A. Yeh, Chen Chen, Teck-Yian Lim, Mark Hasegawa-Johnson, and Minh N. Do. Semantic image inpainting with perceptual and contextual losses. *CVPR*, pages 5505–5514. 3

[30] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. *arXiv preprint arXiv:1806.03589*, 2018. 2

[31] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S. Huang. Generative image inpainting with con-

textual attention. *CVPR*, pages 5505–5514, 2018. 1, 2, 3, 6

[32] Y. Zhang, J. Xiao, J. Hays, and P. Tan. Framebreak: Dramatic image extrapolation by guided shift-maps. *CVPR*, pages 1171–1178, 2013. 3

[33] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *ICCV*, pages 2223–2232, 2017. 1, 2