Foreground-aware Image Inpainting

# Introduction

The paper addresses the problem of filling holes of an image. It describes the usually utilised approaches to solve this problem, which are not very satisfactory when dealing with overlapping holes or with holes overlapping foreground objects. Conventional inpainting methods typically fill missing pixels by matching and pasting patches based on low level features such as Mean Square Difference of RGB values or SIFT descriptors.

To this end, the authors proposed a foreground-aware image inpainting system that explicitly incorporates the foreground object knowledge into the training process. Specifically, their model first detects a foreground contour of the corrupted image, and then completes the missing contours of the foreground objects with a contour completion module. The completed contour along with the input image are then fed to the image completion module as guidance to predict contents in holes. Basically, they did disentangle explicitly the structure inference and image completion to address the challenging scenarios described above, proposed a contour completion module trained explicitly to guide image completion, adopted curriculum training on both the contour and the image completion modules to help with the integration.

# Related work

Image inpainting approaches can be roughly divided into two categories: traditional methods based on pixel propagation or patch matching, and recent methods based on deep neural network training. Traditional methods fill in holes by propagating the neighborhood appearance based on techniques like isophote direction field. These methods are quite effective for small or narrow holes, but when the holes are large or the textures vary heavily, they often generate significant visual artifacts.

Recently, learning based inpainting methods have significantly improved inpainting results by learning semantics from large scale dataset. These methods typically train a convolutional neural network as a mapping function from a corrupted image to a completed one end-to-end. A significant advantage of these methods over the non-learning ones is the ability to learn and understand semantics of images for inpainting, which is especially important in cases of complex scenes, faces, objects and many others.

# Proposed solution’s approach

The solution proposed by the authors is composed of a cascade of three modules: incomplete contour detection module, contour completion module and image completion module. The system automatically detects the contour of the incomplete image using the contour detection module. Then it uses the contour completion module to predict the missing parts of the contour. Finally, it inputs both the incomplete image and the completed contour to the image completion module to predict the final inpainted image.

## Data acquisition and hole generation

### Dataset and contour mask generation

Existing datasets for image inpainting do not require any annotations, and training data pairs (image with hole and the ground-truth image) are typically constructed by generating random masks on the original images and by setting the original pixel values under the masks as the ground truth. The proposed framework for foreground-aware image inpainting requires to train a contour completion module and infer the contour automatically, so the authors needed a training dataset with labeled contours. To do this, they used salient object segmentation datasets as an alternative. The dataset is quite diverse in content, containing a large variety of objects, including animals, plants, persons, faces, buildings, streets and so on. The relative size of objects in each image has a large variance, making the dataset quite challenging. The authors split all the samples into 12,609 training images and 3,153 testing images. They then applied the Sobel edge operator on the segmentation mask to obtain the contours of the salient objects. Specifically, they first obtain the filtered mask by applying the Sobel operator: , where and are the vertical and horizontal derivative approximations of the image, respectively. Then they binarize the filtered mask with a simple threshold and obtain the final binary contour as the ground-truth contour of the original image.

### Hole mask sampling

In order to simulate the real world inputs and learn a practical model, we draw holes on each image with arbitrary shapes randomly with a brush. We generate two types of holes: 1). arbitrarily-shaped holes that can appear in any region of the input image. Under this setting, holes have a probability of overlapping with the foreground objects. This scenario is designed to handle the situations where unwanted objects are inside the foreground objects or partially occlude the salient objects; 2). arbitrarily-shaped holes that are restricted so that they have no overlap with the foreground objects. These types of holes are generated to simulate the situation where the unwanted regions or distracting objects are behind the salient objects. To deal with the second situation, we first randomly generate arbitrarily-shaped holes, then we remove the parts of holes that have overlap with the saliency objects.

### Contour detection

During the inference stage, we do not have a contour mask of the input image. We therefore use DeepCut to detect the saliency objects in the image automatically.

## Contour completion module

### Architecture

The contour completion module is composed of a generator and a discriminator. The generator is a cascade of a coarse network and a refinement network. For training, instead of using predicted contours, we extract a clean incomplete contour of the foreground objects directly from the ground-truth contour with the hole mask , i.e.,. Then we input the incomplete image, the incomplete contour image, and the hole mask into our coarse network, which outputs a coarse complete contour . The coarse network is an encoder-decoder network with several convolutional and dilated convolutional layers. The coarse contour map is a rough estimate of the missing contours. The predicted contours around the holes can be blurry and cannot be used as an effective guidance for the image completion module.

To infer a more accurate contour, we adopt the refinement network which takes the coarse contour as input, and output a cleaner and more precise contour . The refinement network has a similar architecture as the coarse network, except that we use a contextual attention layer, to explicitly attend on global feature patches while inferring the missing values. Note that the pixel value of the predicted contour ranges from 0 to 1, indicating the probability that the pixel to be on the actual contour.

The refined contour is then fed to the contour discriminator for adversarial training. The contour discriminator is a fully convolutional PatchGAN discriminator that outputs a score map instead of a single score, so as to tell the realism of different local regions of the generated contour mask.

### Curriculum training

The training tends to fail if both the content loss and the adversarial loss are applied simultaneously even though the weights between the two types of losses are carefully adjusted. To avoid the issue, the authors used curriculum learning to gradually train the model. In the first stage, the contour completion module is required only to output a rough contour, thus they only trained the model with the content loss. Then in the second stage, they fine-tuned the pre-trained network with the adversarial loss, but with a 5 very small weight compared to the content loss, i.e., 0.01 : 1 to avoid training failure due to the instability of the GAN loss for contour prediction. In the third stage, they fine-tuned the whole contour completion module with the weight of adversarial loss and the weight of content loss to be 1:1.

## Image completion module

### Architecture

The module takes as input an incomplete image , the completed contour and the hole mask as inputs, and outputs the completed image . It shares a similar architecture as the contour completion module. The generator of our image completion module also contains a coarse network and a refinement network. The coarse network outputs a coarsely completed image, which can be blurry with missing details. Then the refinement network takes the coarse image as input, and generates a more accurate result.

The discriminator takes the generated image/ground-truth image along with the hole mask indicating the location of the missing pixels as inputs, and tells whether the input pair is real or fake. Similar to the contour completion module, this module uses a PatchGAN structure and a hinge adversarial loss to train the model.

### Training

This module is first pre-trained on the large-scale places2 dataset without the extra channel for the contour map, then fine-tuned on the saliency dataset with the guidance from the output of the contour completion module. Since the pre-trained model takes different inputs compared to the model trained on the saliency dataset, the authors chose to save all parameters for all layers, except the first one, which has been randomly initialized with parameters from this dataset. To stabilize the training, a similar curriculum training strategy as the training of the contour completion module was used in this module as well.

# Experiments

## Implementation details

The modules are trained with Adam as optimizer, a batch size of 64 and a learning rate of 0.0002.

## Comparison with state-of-the-arts methods

The authors performed analysis of their work against GatedConv, PartialConv, ContextAttention, Global&Local and PatchMatch. They also compared with a version of GatedConv fine-tuned on the dataset chosen for this paper

### Quantitative evaluation

The authors performed this evaluation using 5 metrics: L1 loss, L2 loss, PSNR, SSIM and user preference. On all metrics the solution came on top when compared with the others, with a score of 731 on user preference out of 1099 people that evaluated the quality of the output in comparison with the other methods.

### Qualitative evaluation

The main point on this evaluation is that the proposed solution not only fills holes, but also creates realistic resulting images, as opposed to the other methods, which often create details that aren’t realistic.

# Conclusion

The authors came to the conclusion that the proposed solution is so far the best in filling holes with foreground information, outperforming other state-of-the-art solution both quantitatively and qualitatively. This shows that using structures to indicate the foregrounds and backgrounds of the input image, then explicitly guide the completion of the image is a promising direction for inpainting tasks.