Image Inpainting

• Image inpainting is the task of accurately filling in a removed part of an image with suitable imagery that blends in with the rest of the image.
• Our model combines a GAN with partial convolutions, which is a convolutional method that incrementally updates both the image and the mask.
• This allows the network to learn the broader context of the image and accurately inpaint missing parts and detailed features.

Dataset

Our model is trained on a subset of the dataset Places365. It consists of a large set of images displaying cities, buildings, parks among other things. Of the over 8 million total images, we use a subset of 30,000 images for training and 1,000 for testing.

Pre-processing

• The images are scaled down to 128x128x3 (RGB)
• The masked regions are replaced by the mean value
• All pixel values are normalized by division of 255

Method & Model

Key features of our architecture:

• Encoder-decoder: The image is embedded into a lower dimensional representation (encoding) and then reconstructed into the high-dimensional output space (decoding) [1].
• U-Net architecture with skip connections from encoding to decoding layers so not all the information has to go through the encoder-decoder bottleneck [2].
• Local and global discriminators to improve local consistency around the inpainted regions as suggested by Isola [3].
• Partial convolutions that take into account the mask to avoid conditioning on arbitrary placeholder values in the masked regions as proposed by [4].

Loss Functions

\[ \mathcal{L}_{\text{local}} = \frac{1}{n} \sum_{i=1}^{n} (I_{\text{org}} - I_{\text{i}})^2 \]
\[ \mathcal{L}_{\text{mask}} = \frac{1}{n} \sum_{i=1}^{n} (I_{\text{org}} - I_{\text{m}})^2 \]
\[ \mathcal{L}_{\text{edge}} = \mathcal{L}_{\text{mask}} + \mathcal{L}_{\text{edge}} \]
\[ \mathcal{L}_{\text{adv}} = -\log(D(I_{\text{org}})) \]
\[ \mathcal{L}_{\text{gen}} = \log(D(I_{\text{i}})) \]
\[ \mathcal{L}_{\text{cons}} = -\log(D(G(I_{\text{m}}))) \]

where \( I_{\text{org}} \) is the ground truth, \( I_{\text{m}} \) the mask and \( I_{\text{i}} \) the masked input image.

Partial convolutional layers comprise two steps:

1. A masked and renormalized convolution step where the convolution \( \mathcal{C} \)
\[ \mathcal{C} = \text{max} \left( \mathcal{W} \ast \mathbf{X}, \mathbf{M} \right) + \mathbf{b} \]
\[ \text{if } \mathbf{1} \cdot \mathbf{M} > 0, \]
\[ \text{else} \]

2. A mask-update step where the mask value for the given convolution is expressed as:
\[ m_{x,y} = \begin{cases} 1 & \text{if } \mathbf{1} \cdot \mathbf{M} > 0, \\ 0 & \text{else} \end{cases} \]

Training in three phases

Phase 1: use only weighted MSE loss \( \mathcal{L}_{\text{local}} \) for 25,000 iterations
Phase 2: train only discriminators on \( \mathcal{L}_{\text{adv}} \) for 6,000 iterations
Phase 3: train both generator and discriminators for 14,000 iterations

Results

Significance of adversarial loss
Training only on mean squared error (MSE) produces blurry results

Significance of partial convolutions
Reduces color discrepancies and blur increases checkerboard artifacts

Significance of local discriminators
Only slight differences observable tend to produce finer details

Conclusion

Main results

• To the best of our knowledge, we are the first to train a GAN with partial convolutions.
• Adversarial training is absolutely crucial to reproduce the finer details of an image.
• Good results can be obtained with significantly smaller networks than state-of-the-art approaches.
• Partial convolutions improve the results compared to typical convolutions.

Future work

• With more computational resources, the model could have been trained to inpaint irregular shapes.
• Since partial convolutions successively fill the holes from the edges inward, the size of the inpainted region is limited by the depth of the network.
• Future work could train a deeper network on irregular masks to perform well on larger and randomly selected masks.
References


