The 4th International Conference on Computing and Data Science (CONF-CDS 2022)
DOI: 10.54254/2755-2721/2/20220623

Remover: Region-Based Inpainting Algorithm

Huifeixin Chen[1,*] , Yining Xu[2]

[1]College of Information and Intelligence, Hunan Agricultural University, Hunan 410128, China
[2]School of Control Science and Engineering, Shandong University, Shandong 250002, China

chenhui123@stu.hunau.edu.cn

Abstract. The existing method does not seem to be a killer application that combines image segmentation and inpainting to do image processing tasks for ordinary people. Therefore, we propose a region-based inpainting method, namely Remover. Remover is a method that can analyze the content of images, perform automatic image segmentation tasks with or without manual intervention, and inpainting the segmented part to achieve unwanted objects appearing to have been removed from an image without affecting the content of the image. With the help of opensource code and technical support, Remover stands as an Ubuntu desktop application. The code is under development and will be available at https://github.com/WPCJATH/remover soon.

Keywords: image inpainting, computer vision, machine learning, detectron2, edgeconnect.

1. Introduction

Traditional graphics and visual research methods are mainly based on mathematical and physical methods. However, with the remarkable development of deep learning[1] in the field of vision in reent years, the frontier of visual field research has been occupied by deep learning. Deep learning technology[1-4] has made significant progress in picture inpainting in recent years. Image inpainting requires the algorithm to complete the missing area of the image and then restore it according to the information in the image itself or the image library, making the restored image look very natural and difficult to distinguish from the entire image. The term "image inpainting" refers to filling in the missing or damaged sections of a reconstructed picture or video. In many picture editing projects, this is a crucial step. It can be used to fill in the gaps left by deleting undesired elements from an image or repairing damaged photographs, for example. Contour generation and image mosaic are the two processes of picture inpainting.

On the other hand, these algorithms cannot recreate the sensible structure of the missing parts in the image and invariably provide overly smooth or fuzzy results. EdgeConnect is a brand-new image patching system. It has a special patching effect and more delicate features in the filled regions.

EdgeConnect[5] connects the image patching network to the edge generator. As and a priori outcome, the edge generator creates an edge imaginary map of irregular missing areas, which is then used by the image patching network to fill in the missing areas based on this edge imaginary map. The EdgeConnect model is tested end-to-end on a publicly available data set, and the findings reveal that

DOI: 10.54254/2755-2721/2/20220623

EdgeConnect outperforms competing methods at this point. In recent years, deep learning-based picture restoration technology has demonstrated promising results. It can fill in the image's gaps with semantically valid context-aware data. Although these learning-based[6] methods are far more effective than earlier technologies in capturing sophisticated information, they can only analyze low-resolution input due to memory restrictions and training challenges. The mended portions may seem blurry even for somewhat bigger photos, and restoration results that do not meet expectations will become more common. High-Resolution[7] Image Inpainting presents a convolution neural network technique that simultaneously optimizes image content and texture restrictions. By matching and altering the patches with the deep classification net-work's most similar middle-level feature correlation, this method not only preserves the context structure but also generates high-frequency details. This approach achieves the highest repair accuracy on the ImageNet[8] and Paris Streetview datasets. The results reveal that, especially for high-resolution photographs, this method gives better and more consistent results than earlier methods. Most CNN-based image restoration approaches do not distinguish between effective pixels and holes, limiting their ability to deal with irregular holes and increasing the likelihood of color variations and hazy restoration outcomes. Some people recommend partial convolution to fix this problem, but it only examines the forward mask update and employs artificial operations to normalize the features. Learnable Bidirectional Attention Maps offer an end-to-end learnable attention graph module for learning feature renormalization and mask updating. This program can easily handle the propagation of irregular holes and convolution layers. A learnable reverse attention graph is also added, allowing the unet decoder to concentrate on filling irregular holes rather than reconstructing holes and known areas, resulting in a learnable bidirectional attention graph. According to qualitative and quantitative tests, the Learnable Bidirectional Attention Maps method outperforms existing technologies to produce more transparent, more coherent, and visually plausible repair results.

In this paper, the primary purpose of this project is to use the instance segmentation technique to extract all of the items in a picture and produce masks, labels, and confidence ratings for each object. The mask will give the input image a translucent tint, and the user will see the label and confidence score as text, allowing them to choose which object to remove. At the same time, the user may use the operation interface to draw the region on the original design that has to be erased and update and improve the area of the recognized object. Finally, execute the repair algorithm, fill in the image's removal zone, and the remover will generate the repair result.

2. Method

We will detail the framework of our method. Our method contains three components: image instance segmentation (IIS), edge detection (ED), and region-based inpainting (RI). IIS takes the images as input and outputs the segmentation mask for each object, and ED provides the edge contour of the ROI and then sends it to the RI for image inpainting.

2.1. Image Instance Segmentation Using Faster RCNN

From R-CNN[9], Fast R-CNN to Faster R-CNN, the speed of image feature extraction by computer is constantly accelerated, and the efficiency is constantly improved. In this paper, we mainly adopt the Faster R-CNN model structure. Based on Fast R-CNN, this structure introduces the RPN network to replace the original selective search method to produce Anchor boxes and shares the CNN of Anchor boxes with the CNN of target detection, creatively reducing the original 2000 boxes to around 300 and improving their quality. The faster R-CNN network structure mainly realizes the generation of the input image anchor boxes, feature extraction, classification, and edge refinement. Faster R-CNN is mainly composed of three modules: the Backbone Network, Region Proposal Network, and ROL Heads. The Backbone Network extracts feature maps from the input image at different scales. The Region Proposal Network detects object regions from multiscale features. Box Head crops and warps feature maps using proposal boxes into multiple fixed size features and obtain finetuned box locations and classification results via fully connected layers.

DOI: 10.54254/2755-2721/2/20220623

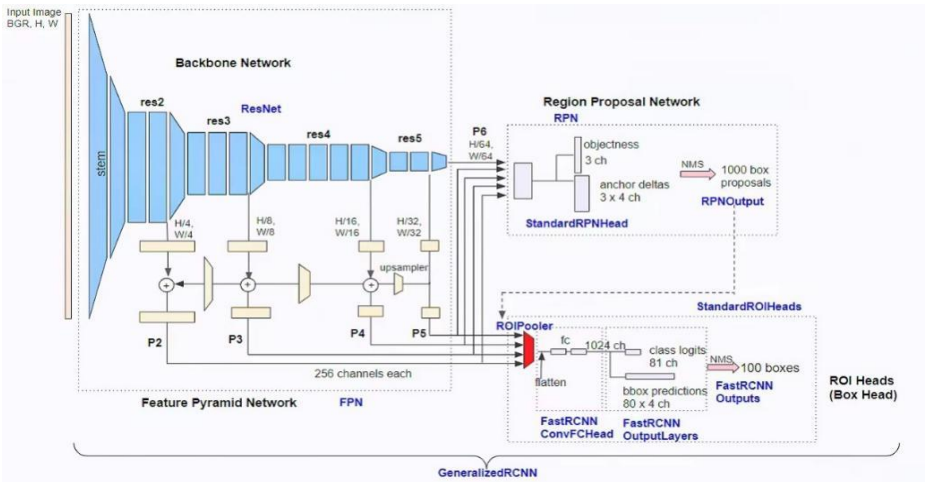

Figure 1. Architecture of Base-RCNN-FPN of Detectron2[10].

2.2. Edge Detection Using Canny Operator

It takes five steps for a Canny operator to extract edge information:

1.If the input image is color, it should be gray scaled.

2.The following operations are performed on the image successively: the image is smoothed by a Gaussian filter; the gradient amplitude and orientation are calculated by firstorder partial differential finite difference, and the gradient amplitude is not significantly suppressed.

3.The double threshold algorithm detects and connects the edges.

The effect of Canny edge detection is remarkable. The pseudo-edge caused by noise is significantly suppressed compared with the conventional gradient algorithm. This algorithm refines the edge, which is easy for subsequent processing. The Canny algorithm can also achieve good results for images with low contrast by adjusting parameters.

2.3. Image Inpainting Using EdgeConnect

EdgeConnect, as a new inpainting method, carries out more detailed image restoration. The EdgeConnect model outlines the image's edge lines and then completes the coloring work to reproduce the filled area better. Concretely, the two-stage adversarial model EdgeConnect comprises an Edge Generator and an image completion network. The Edge Generator only focuses on generating imaginary edge contours in the missing region. Furthermore, the image completion network estimates the RGB pixel value of the missing area using the imaginary contour image and the incomplete input image.

DOI: 10.54254/2755-2721/2/20220623

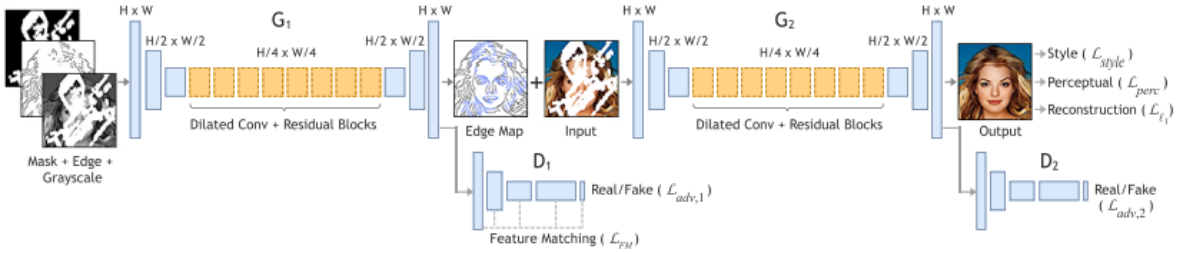

Figure 2. Architecture of EdgeConnect.

Edge Generator. The genuine picture is recorded as $\mathbf{I}_t$ in the part of the edge generator, and its edge image and gray-scale image are represented by $\mathbf{C}_{gt}$ and $\mathbf{I}_{\text{gray}}$, respectively. As input, we employ the grayscale image mask $\tilde{\mathbf{I}}_{\text{gray}} = \mathbf{I}_{\text{gray}} \odot (\mathbf{1} - \mathbf{M})$, its edge mapping $\tilde{\mathbf{C}}_{\boldsymbol{gt}} = \mathbf{C}_{\boldsymbol{gt}} \odot (\mathbf{1} - \mathbf{M})$, and the image mask $\mathbf{M}$ as the preconditioner (Make a 1 for the missing region and a 0 for the background image.). Adama products are represented here. The edge map of the mask region is predicted by the generator.

$$\mathbf{C}_{\text{pred}} = \boldsymbol{G_1}\left(\tilde{\mathbf{I}}_{\text{gray}}, \tilde{\mathbf{C}}_{\boldsymbol{gt}}, \mathbf{M}\right) \tag{1}$$

The loss function is created to train the countermeasure network to obtain the contour generator edge generator. $\mathcal{L}_{adv,1}$ is adversarial loss, $\mathcal{L}_{FM}$ is feature map loss.

$$\begin{aligned}\boldsymbol{\mathcal{L}_{adv,1}} = \mathbb{E}_{(\boldsymbol{C}_{\text{gt}}, \boldsymbol{I}_{\text{gray}})} \mathbf{log}\left[\boldsymbol{D_1}\left(\boldsymbol{C_{gt}}, \boldsymbol{I}_{\text{gray}}\right)\right] \\ + \mathbb{E}_{\boldsymbol{I}_{\text{gray}}} \mathbf{log}\left[\mathbf{1} - \boldsymbol{D_1}\left(\boldsymbol{C}_{\text{pred}}, \boldsymbol{I}_{\text{gray}}\right)\right]\end{aligned} \tag{2}$$

$$\boldsymbol{\mathcal{L}_{FM}} = \mathbb{E}\left[\sum_{\boldsymbol{i=1}}^{\boldsymbol{L}} \frac{\mathbf{1}}{\boldsymbol{N_i}} \parallel \boldsymbol{D_1^{(i)}}\left(\boldsymbol{C_{gt}} - \boldsymbol{D_1^{(i)}}\left(\boldsymbol{C_{pred}}\right) \parallel_1\right]\right] \tag{3}$$

The input image is judged using the pre-trained VGG network. PatchGAN is a method that is comparable to this one. We cannot immediately use the results of VGG because it is not a network trained to extract the contour edge of an image. The discriminator's last convolution layer is represented by $\mathcal{L}$. The activation result in the discriminator's $i'th$ layer is N_i.

The contour discriminator, which combines the confrontation loss and the feature matching loss, distinguishes the contour image:

$$\begin{gathered}\min_{G_1}\max_{D_1}\mathcal{L}_{G_1} = \min_{G_1}\left(\lambda_{adv,1}\max_{D_1}\left(\mathcal{L}_{adv,1}\right) + \lambda_{FM}\mathcal{L}_{FM}\right) \\ \lambda_{adv,1} = 1, \lambda_{FM} = 10\end{gathered} \tag{4}$$

Image Completion Network. The incomplete color image $\tilde{\boldsymbol{I}}_{\text{gray}} = \boldsymbol{I}_{\text{gray}} \odot (\mathbf{1} - \boldsymbol{M})$ is sent into the image completion network, which is then conditioned using a composite edge map C_{comp}. The composite edge $\boldsymbol{C_{comp}} = \boldsymbol{C_{gt}} \odot (\mathbf{1} - \boldsymbol{M}) + \boldsymbol{C_{gred}} \odot \boldsymbol{M}$ combines the background region of ground truth edges with the produced edges in the corrupted zone from the previous stage to create the composite edge map. The network provides a color picture of the exact resolution as the input image $\mathbf{I_{pred}}$, with missing parts filled in:

$$I_{pred} = G_2\left(\tilde{I}_{\text{gray}}, C_{comp}\right) \tag{5}$$

The loss function is created to train the countermeasure network to obtain the contour generator edge generator. $\mathcal{L}_{adv,2}$ is adversarial loss, $\mathcal{L}_{prec}$ is perceptual loss, and $\mathcal{L}_{style}$ is style loss.

$$\begin{aligned}\mathcal{L}_{\text{adv,2}} = \mathbb{E}_{(\text{I}_{\text{gt}}, \text{C}_{\text{comp}})} \text{log}\left[\text{D}_2\left(\text{I}_{\text{gt}}, \text{C}_{\text{comp}}\right)\right] \\ + \mathbb{E}_{\text{C}_{\text{comp}}} \text{log}\left[1 - \text{D}_2\left(\text{I}_{\text{pred}}, \text{C}_{\text{comp}}\right)\right]\end{aligned} \tag{6}$$

DOI: 10.54254/2755-2721/2/20220623

$$\mathcal{L}_{\text{prec}} = \mathbb{E}\left[\sum_{\text{i}}^{\text{L}} \frac{1}{\text{N}_\text{i}} \| \phi_1^{(\text{i})}(\text{I}_{\text{gt}} - \phi_1^{(\text{i})}(\text{I}_{\text{pred}}) \|_1\right] \tag{7}$$

$$\mathcal{L}_{\text{style}} = \mathbb{E}_{\text{j}}\left[\| \text{G}_{\text{j}}^{\phi}\left(\text{I}_{\text{pred}}\right) - \text{G}_{\text{j}}^{\phi}(\text{I}_{\text{gt}}) \|_1\right] \tag{8}$$

To separate the contour picture, a contour discriminator that includes the absolute value norm (L1 distance ℓ_1), resistance loss, perception loss, and style loss is used:

$$\min_{G_1}\max_{D_1}\mathcal{L}_{G_1} = \left(\lambda_{\ell_1}\mathcal{L}_{\ell_1} + \lambda_{adv,2}\max_{D_1}\left(\mathcal{L}_{adv,2}\right) + \lambda_p\mathcal{L}_{prec} + \lambda_{style}\mathcal{L}_{style}\right) \tag{9}$$
$$\lambda_{\ell_1} = 1, \lambda_{adv,2} = \lambda_p = 0.1, \lambda_{style} = 250$$

3. Results and Discussion

3.1. Dataset Description

The performance of image categorization algorithms has traditionally been measured against ImageNet datasets. Image segmentation is performed using the ImageNet data set in this study. The ImageNet dataset is a computer vision dataset developed at Stanford University under the direction of Professor Li. There are 14197122 images and 21841 synset indexes in the data set. In the WordNet hierarchy, Synset is a node. It also includes a list of synonyms.

Figure 3. Sample of ImageNet.

The ImageNet dataset is an extensive picture collection created to help advance computer image recognition technologies. The ImageNet dataset had over 10,000 photos in 2016, and each image was carefully categorized. The photos in the ImageNet collection span the vast majority of image categories people will see daily. ImageNet started as a data collection containing over a million

DOI: 10.54254/2755-2721/2/20220623

photos. It has a variety of pictures, as indicated in the diagram below, and each image is labeled (class alias).

People usually use the places2 dataset in the image inpainting section. The places2 dataset has about 10 million photos, with over 400 different scene classifications. Each category in the dataset comprises 5000 to 30000 training photos, which corresponds to scene frequency in the current world.

The Places2 dataset primarily consists of three datasets:

The entire set of the Places2 database is the Places365 standard. In the places365 standard for training places365 CNN, there are 1.8 million training photos from 365 scene categories. The validation set has 50 images per category, whereas the test set contains 900 images per category.

Places365 Challenge 2016 is a competition for the 2016 places365 challenge, including 8 million training photos (including places365 standard training data). The verification and test sets are identical to those used in the places 365 standards.

Places-Extra69: In addition to the 365 scene categories released in Places365, we have released image data for an additional 69 scene categories (totaling 434 in the Places Database) as Places-Extra69.

3.2. Comparison Results

The following is the analysis of the image restoration results.

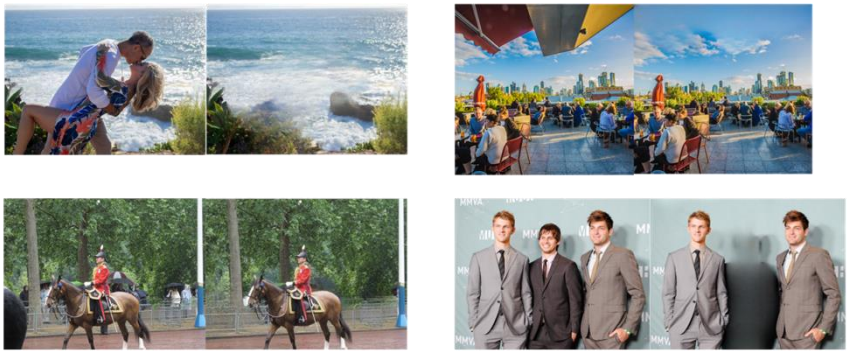

Figure 4. Visualization of Remover.

The image processing in the upper left corner is to remove the person from the image. By comparing the images before and after processing, it can be concluded that the purpose of image inpainting of the program is achieved, the image is consistent with cognition, and the color is reasonable. However, EdgeConnect is not enough to correctly handle the object approaching the image boundary, and there is an image blur problem after the image processing.

The two images on the upper right are before and after the awning was removed. Before and after image processing, the blank of the removed part of the awning uses clouds to fill, which is logical and cognitive. Nevertheless, the details can still be improved. For example, the shadows cast by the awning must be removed simultaneously during the image processing.

Machine repair the image in the lower left corner successfully, basically achieving the purpose of removing the tourists in the background. When smearing the selection, the horse and man's boundaries in the foreground were identified more successfully, and the restored image was more realistic than the previous photos.

As seen from the bottom right corner of the fix, there is still much room for improvement in handling the problem of objects approaching the boundaries of other objects. The user tries to remove the central figure through the program, and it can be seen from the processing results that part of the middle figure is removed. However, it does not achieve a good removal effect, and even part of the outline is not identified.

DOI: 10.54254/2755-2721/2/20220623

4. Conclusion

In this paper, we propose a region-based inpainting algorithm, namely Remover. The Remover combines image instance segmentation with Detetron2, plus manual adjustment. The image after edge detection is fed into EdgeConnect Inpainting for object removal and image restoration. The designers optimized the fixes with sample algorithms and designed a user-friendly interface to improve the availability of the Remover. While Remover is currently inadequate to handle all object detection and removal situations, there are still issues such as poor edge connections and partially blurred fill. However, it is compatible in most scenarios, the overall fix idea is logical, and the color filling is reasonable. We state that the proposed Remover has great potential on real-world applications, including mobile photograph and photo editing.

References

[1] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT press.
[2] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. nature, 521(7553), 436-444.
[3] Pouyanfar, S., Sadiq, S., Yan, Y., Tian, H., Tao, Y., Reyes, M. P., ... & Iyengar, S. S. (2018). A survey on deep learning: Algorithms, techniques, and applications. ACM Computing Surveys (CSUR), 51(5), 1-36.
[4] Shrestha, A., & Mahmood, A. (2019). Review of deep learning algorithms and architectures. IEEE Access, 7, 53040-53065.
[5] Nazeri, K., Ng, E., Joseph, T., Qureshi, F. Z., & Ebrahimi, M. (2019, January 1). EdgeConnect: Generative image inpainting with adversarial edge learning. ArXiv.Org.
[6] Xie, C., Liu, S., Li, C., Cheng, M.-M., Zuo, W., Liu, X., Wen, S., & Ding, E. (2019, September 3). Image inpainting with learnable bidirectional attention maps. ArXiv.Org.
[7] Yang, C., Lu, X., Lin, Z., Shechtman, E., Wang, O., & Li, H. (2016, November 30). High-Resolution Image Inpainting using Multi-Scale Neural Patch Synthesis. ArXiv.Org.
[8] Deng, J., Dong, W., Socher, R., Li, L.-J., Kai Li, & Li Fei-Fei. (2009, June). ImageNet: A large-scale hierarchical image database. 2009 IEEE Conference on Computer Vision and Pattern Recognition.
[9] Honda, H. (2022, January 18). Digging into Detectron 2 — part 1 - Hiroto Honda. Medium. https://medium.com/@hirotoschwert/digging-into-detectron-2-47b2e794fabd.
[10] Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(6), 1137–1149.