Proceedings of the 4th International Conference on Computing and Data Science (CONF-CDS 2022)
DOI: 10.54254/2755-2721/2/20220668

Precise Human Removal and Inpainting Using Mask RCNN and LaMa

Xiangzhi Wang

The Hong Kong Polytechnic University, Department of Computing, HKSAR

19082878d@connect.polyu.hk

Abstract. Sometimes people are not supposed to be in a photo for various purposes, but this is usually unavoidable. Therefore, in the post-processing of the image, it can be solved by removing people from the picture without affecting the coherence and naturalness of the object and background in the photo. We propose a human removal method based on image instance segmentation and image inpainting. Firstly, we send an image into the image instance segmentation algorithm to obtain a mask covering the unwanted parts of the picture. Then we do dilatation on this mask to expand the mask region. Finally, the inpainting algorithm will take the image and the processed mask and produce an inpainted image with no human and feel natural.

Keywords: image inpainting, image instance segmentation, privacy protection, photo enhancement.

1. Introduction

Maybe people do not like being in the photos released on the Internet. Perhaps the picture owner does not want pedestrians or tourists to appear in the scene. The crowds may make the image's content chaotic if the people in the image could be removed and the obscured scene or background could be recovered automatically, significantly improving the photos' pre - and post-processing efficiency. We propose a human removal method based on image instance segmentation and image inpainting. Firstly, we send an image into the image instance segmentation algorithm to obtain a mask covering the unwanted parts of the image. Then we do dilatation on this mask to expand the mask region. Finally, the inpainting algorithm will take the image and the processed mask and produce an inpainted image with no human and feel natural. This method would apply to privacy protection, image restoration, photo enhancement, etc.

1.1 Image Instance Segmentation

Image instance segmentation orientates various object instances that appear in images to predict classes of the objects and assign the mask to the objects at pixel-level within images [1, 2, 3, 4, 5]. It contributes to the development of automatic drive, image process, privacy protection, etc. Image instance segmentation treats the "stuff" and "thing" objects differently. There is no instance concept for "stuff" objects, but different "thing" objects of the same category (label or class) will be separated during instance segmentation [2, 6], which is where semantic segmentation [1, 2, 6, 7], not differentiating "thing" and

DOI: 10.54254/2755-2721/2/20220668

"stuff" objects, differs from instance segmentation [2, 6]. Fig 1 illustrates the difference between semantic segmentation and instance segmentation.

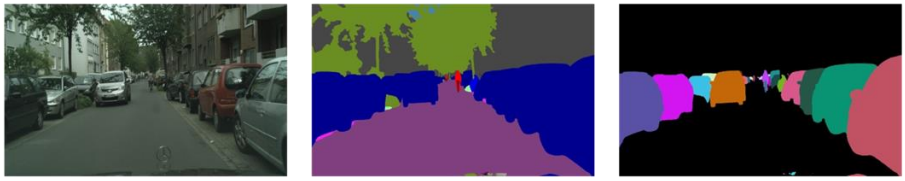


Figure 1. Image (left), semantic segmentation (middle), instance segmentation ("stuff" objects are ignored) (right) [6].

1.2 Image Inpainting

In the context of modern times, image inpainting refers to filling or recovering the missed, destroyed, or unwanted parts of the image and making the image as natural as possible (seemed no modifications or processes had been done) by taking advantage of computer power [8-13]. Artwork, selfie, photograph, etc., as long as the image is in electronic form, it can be processed by image inpainting. Image inpainting could be divided into sequential-based, CNN-based, and generative adversarial network (GAN) [14] based [9]. Here, GAN-based methods will be focused on. Two feed-forward networks would be used as the generator and discriminator for the GAN-based approaches. The generator will be trained for produced a new image that cannot be distinguished from real ones, while the discriminator will be trained to distinguish the generated and real ones. Thus, there was a confrontation between the generator and the discriminator [9]. A coarse-to-fine network architecture consisting of a GAN network producing an initial inpainting result and another building a refined inpainted output as the final result is adopted [9]. GAN-based image inpainting techs [10-13] usually take an image along with its mask (a binary image to indicate which pixels will be removed or recovered and which are not) and produce the inpainted image during inference tasks, as shown at Fig. 2.

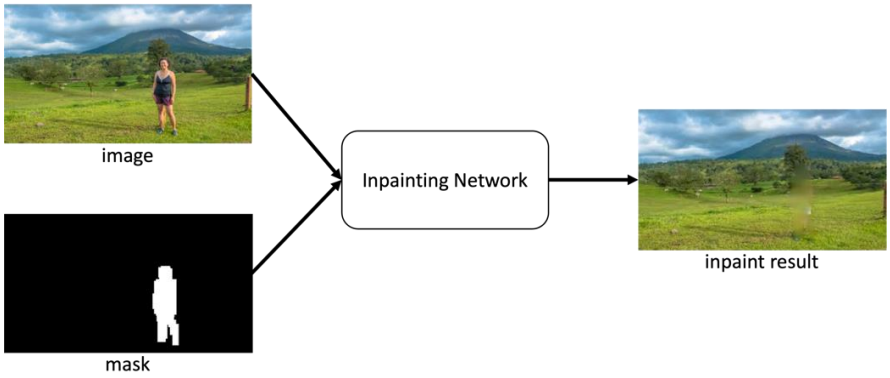


Figure 2. The input and output of GAN-based image inpainting.

1.3 Related Work

RCNN Variant. As one of the earliest techniques adopts convolutional neural networks (CNNs) in the field of instance segmentation [1], Region with CNN features (RCNN) [15] achieves a much higher mean average precision (mAP) on PASCAL VOC 2012 than the 2nd best method [15]. 2000 region proposals would be generated with the help of selective search [16] and put into a CNN to do feature extraction tasks. Then, each feature map of the region proposals goes through a support vector machine (SVM) classifier and bounding box (BBox) regressor, respectively, to produce BBoxes and their corresponding classes [16]. However, the training and testing procedures of RCNN are relatively slow due to complex and massive computations [1]. FastRCNN [17] and Faster RCNN [18] are built to address such

DOI: 10.54254/2755-2721/2/20220668

problems. Compared to RCNN, Fast RCNN directly puts the image with its region proposals (also produced by selective search) into a CNN instead of the 2000 region proposals (by using selective search) and utilizes the region of interest (RoI) pooling layer with the support of SPPNets [19], to generate the RoI feature vector. The SVM classifier is replaced by a fully connected (fc) layer with SoftMax [17]. A region proposal neural (RPN) network [18] is introduced and replaces the selective search technical to achieve better performance and higher accuracy at FasterRCNN [18]. The feature maps go through the RPN to get proposals. With the feature maps (shared), the proposals go through the RoI pooling layer and box-regressor/box-classifier to output the results [18].

Path Aggregation Network (PANet). Besides adopting a feature pyramid network (FPN) [3] as the backbone, an augmented bottom-up path architecture is introduced to achieve an easier propagation effect. Adaptive feature pooling is also introduced to generate proposals. Masks are generated based on the fc fusion technical. PANet wins the COCO [20] 2017 Instance Segmentation Task Challenge in the first place. According to [1, 21].

YOLACT. achieves the real-time instance segmentation tasks and maintains relatively feasible precision. Like MaskRCNN [5] (covered by supplementary materials), YOLACT adjusts in the same place as FasterRCNN, but it branches into two paths, i.e., parallel tasks. One path uses fully convolutional networks (FCN) [22] to draw "prototype masks" without knowing the existence of any instances, and the other one uses non-maximum suppression (NMS) to obtain "mask coefficients". After the linear combination of the output of the two paths, i.e., the "prototype masks" and "mask coefficients", clipping and threshold are carried out [4].

Deep Fill. A coarse-to-fine network is proposed by [10] with dilated convolutional layers widely involved. Since the Yu. elta. [10] claim that for the holes to be filled during inpainting. The boundary regions turned out to be more important than the center. The reconstruction losses used by both coarse and fine networks are weightily calculated. Contextual Attention layers are added into one batch of the fine network learning to do "copy and paste" works to fill the holes. Deep Fill achieves high quantity image inpainting tasks but has limitations on high-resolution images [10].

Edge Connect. Nazeriel ta. [11] also introduced a coarse-to-fine liked network. However, instead of producing the initial coarse result, the coarse network takes the edge map calculated by the canny edge detector [23] based on the mask and image. A predicted and completed edge map will then be outputted and go to the refine network with the original image and its mask to generate the inpainted image. This is also a GAN-based network with two pairs of generators and discriminators for these 2-step networks, respectively similar to Deep Fill, and dilated convolutional and residual blocks are widely used. Edge Connect also has good effect inpainting capabilities, even better than Deep Fill, but also unsuitable for large-scale images [11].

HiFill. A few adjustments on the coarse-to-fine network compared to Deep Fill, making HiFill [12] capable of large-scale images (up to 8K), fast, real-time calculation, and support on "big holes" (up to 25%), i.e., the region on image to be overlaid. The coarse network is consisted of dilated convolution activations and lightweight gated convolution activations (LWGC) in a symmetric pyramid-liked architecture. In contrast, the refine network uses an attention transfer module (ATM) to do weight "copy and paste" tasks. Besides, a mechanism named contextual residual aggregation (CRA) is defined outside the coarse-to-fine network enabling inpainting quality on large-scale images, upgrading the calculation efficiency, and making it possible to train small images and inferencing on large-scale images while ensuring the quality [12].

COCO. Microsoft Common Objects in Context (COCO) [22] is a public dataset designed for advanced object recognition tasks. COCO is a collection of large-scale images that cover daily scenes with everyday objects, i.e., 91 things are included, and the authors of [22] claim that a 4-year-old kid could recognize these objects well. Both iconic object/scene images and non-iconic images are included to solve that recognition systems do not do well on non-iconic objects. For labeling tasks done by COCO, "thing" and "stuff" instances are well distinguished, and the ground-truth instance segmentation for images has been provided. [22] 80k+ train images, 40k+ validation images, and 80k+ test images are provided by the detection challenge of COCO, which is the most relevant to instance segmentation [1].

DOI: 10.54254/2755-2721/2/20220668

COCO has become important and popular due to its large-scale image collection [1] and contributes to instance segmentation tasks, e.g., Mask R-CNN, PANet, and YOLACT all work on COCO. The sample images of COCO are listed in Fig 3.

Figure 3. The sample images of COCO [22].

Places. is an image dataset focused on scenes. It contains 10 million images with 400+ categories, including bridge, hotel room, galley, etc. [24, 25]. Zhou elta. [24] claim that Places is a "quasi-exhaustive" database since it covers nearly all places in the world that humans may encounter with. Images are classified into and labeled with categories strictly during the construction of Places. [25] put forward and proved that object-centric (e.g., COCO [20], ImageNet [26], SUN [27], etc.) datasets and scene-centric datasets have different applicability for CNN models with other purposes. Therefore, in many inpainting techniques (e.g., Deep Fill [10], Edge Connect [11], HiFill [12], etc.) for scene-based inpainting, Places are used for model training, while object-based datasets are used for object detection or segmentation tasks. As the largest scene-centric image database, Places is competitive in training CNNs which require a tremendous amount of data [25]. The sample images of Places are shown in Fig 4.

Figure 4. The sample images of Places [24].

2. MaskRCNN

Besides inheriting the region proposal networks, BBox classifier, and BBox regressor of FasterRCNN [18], MaskRCNN [5] replaces the RoI pooling layer with the RoI alignment layer. Moreover, two versions of FasterRCNN were proposed by [18] and [3], respectively. The original version of FasterRCNN [18] simply takes the final layer of stage 4 from ResNet [28] as the last feature map. In contrast, the feature pyramid network (FPN) [3] version extracts a few final layer outputs from different stages and feeds them into region proposal networks one by one. The authors of [5] argue that the FPN backbone could hugely improve accuracy and spending. The architecture of MaskRCNN-FPN is shown in Fig. 5. The details will be explained below:

DOI: 10.54254/2755-2721/2/20220668

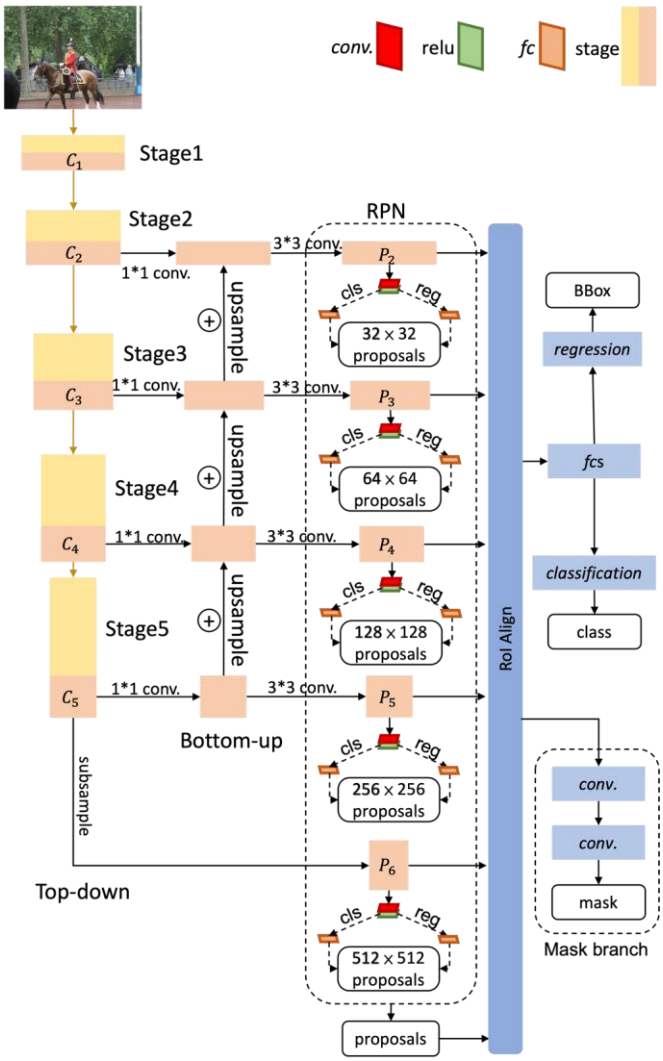


Figure 5. MaskRCNN-FPN.

2.1. Region Proposal Networks

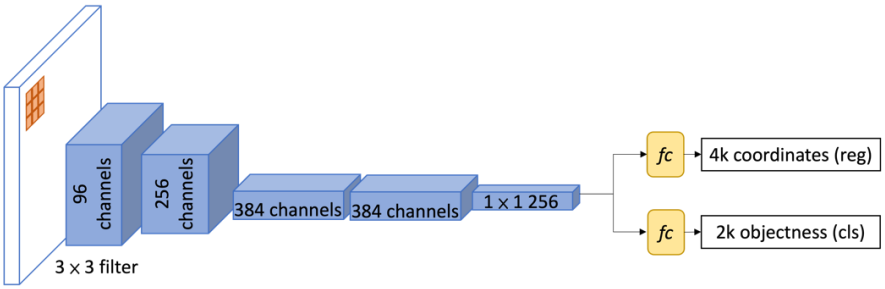


Figure 6. Region Proposal Networks.

Region Proposal Networks (RPN) [18] take feature maps of an image, having gone through an FCN, as input and produce the object proposals in rectangular shapes with scores called “objectness score”. The score measures the extent of whether the proposal corresponds to contains object (foreground) or background. The object here refers to a set of object classes and is covered by the overall framework (i.e., person, horse, car, etc.). The object proposals are generated based on anchors boxes. Anchors are pixels

DOI: 10.54254/2755-2721/2/20220668

picked up by 3×3 kernels slicing at the whole feature maps and regarded as the centric point of the anchor boxes with a defined stride.

Despite the anchors, anchor boxes have properties defined as the scales and aspect ratios, contributing to each box's absolute size and location. The authors of [18] represent three values of scales and another three values of ratios, i.e., each anchor produces nine anchor boxes of different sizes and shapes, as shown in Fig. 7 (left). Note that these two fcs are inside the RPN and have other usages from the BBox regressor and BBox classifier outside the RPN, which will be explained below. The detailed architecture of RPN is shown in Fig. 6.

FPN version RPN has made some changes to the original version. The RPN illustrated above is regarded as a single unit. Each unit will take different feature maps produced by FPN, which can be told in Fig. 5. For different-size feature maps, different-size anchors are taken, e.g., 32×32 anchor boxes are taken from P_2, and 512×512 anchor boxes are taken from P_6. All the proposals will be collected and fed into the RoI Alignment layer.

2.2. RoI Alignment

RoI Pooling. In FastRCNN [18], each RoI proposal and the feature map of the input image would be put into the RoI pooling layer to generate the RoI feature vector. There are defined and fixed values (W,H) coverts the content of RoI, where RoI is described by (x,y,h,w), i.e., the coordinate of the left-top pixel along with the height and width of RoI, into a set of slices with height h/H and width w/W and max pooling will be done on these slices. The (x,y,h,w) values, produced by RPN, and also the height h/H and width w/W of the slices, however, would be floating numbers and will be rounded for further processing, which may lead to "misalignment," i.e., the RoIs is not 100% match with the ground truth BBox and affect the accuracy of the final outputs according to [5]. RoI Alignment is proposed to solve such a problem.

RoI Alignment. Bilinear interpolation [29] is the core of RoI Alignment, which replaces the rounding procedures used in RoI pooling. Given the fixed size of the slices (W,H), there are W×H pixels that should be sampled from the feature map, bilinear interpolation leads to extracting the nearby pixels to be sampled into the slices, and therefore, no rounding would be operated. Fig. 7 (right) illustrates an example of bilinear interpolation in this case, where the blue dashed lines are the feature map with one pixel in each grid and the solid square presents the slices. Each slice would sample 4 pixels from the feature map.

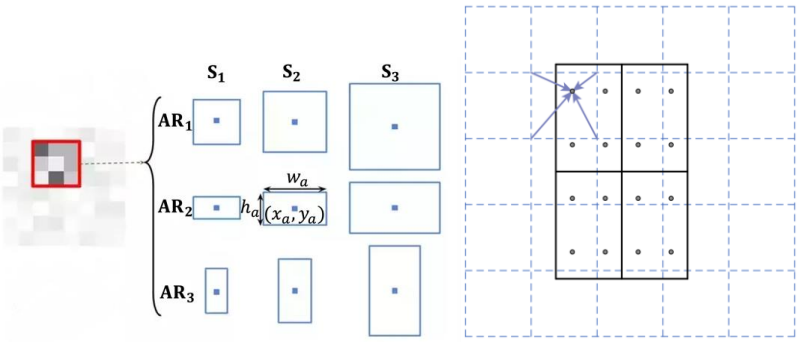


Figure 7. Anchor boxes generation (left) and Bilinear interpolation on RoI Alignment (right) [5].

2.3. Loss Functions

MaskRCNN contains several different networks for different tasks. The multi-task loss on each RoI is [5]:

$$L = L_{class} + L_{bbox} + L_{mask} \tag{1}$$

Class Loss & BBox Loss. L_class and L_box are defined at FastRCNN [17]:

DOI: 10.54254/2755-2721/2/20220668

$$L_{class}(p,u) = -\log p_u \tag{2}$$

$$L_{bbox}(t^u,v) = \begin{cases} \sum_{i\in(x,y,w,h)} smooth_{L1}(t_i^u - v_i), & if\ u > 0 \\ 0, & otherwise \end{cases} \tag{3}$$

$$smooth_{L1}(x) = \begin{cases} 0.5x^2, & if\ |x| < 1 \\ |x| - 0.5, & otherwise \end{cases} \tag{4}$$

For each RoI, $p = (p_0, p_1, \dots, p_k)$ where k is the number of classes covered. p_u $(0 \le u \le k)$ is the probability for current RoI on class u and p_0 is expressly set as the probability of background, i.e., $u = 0$ does not belong to any classes. Similarly, $t^k = (t_x^k, t_y^k, t_w^k, t_h^k)$ on each RoI represents the BBox of the k classes (t^0 is not covered since there is no BBox for background) respectively. In both Eq. 2 and Eq.3, u is the ground-truth class of the current RoI. And v is the ground-truth BBox of the ground truth of u at Eq3. Eq 4. computes the robust L_1 loss, compared to L_2 loss has less sensitivity on outliers, according to [17].

$$L_{mask}(s^u, v') = \frac{1}{m^2}\sum_{i,j} L_{\log}(s_{ij}^u, v_{ij}') \tag{5}$$

$$L_{log}(x,v) = -[v\log(x) + (1-v)\log(1-x)] \tag{6}$$

Mask Loss. The mask network produces k masks corresponding to each class, and for each mask, the resolution is $m \times m$. The mask could be described by $s^k = \begin{bmatrix} s_{11}^k & \cdots & s_{1m}^k \\ \vdots & \ddots & \vdots \\ s_{m1}^k & \cdots & s_{mm}^k \end{bmatrix}$ where k is the number of classes and s_{ij}^k $(0 \le i, < m)$ presents the pixel of class k at position (i,j). Eq. 5 calculates each pixel's mask loss and the average binary entropy loss (Eq 6.). u again is the ground-truth class of current RoI, and v' is the ground-truth mask of u on this RoI.

RPN Loss Function. Different from FasterRCNN [18], adopting the shared parameters, [5] trains the RPN and MaskRCNN separately, and there are no **parameters** to share even though they are sharable. RPN produces 2 sets of values, i.e., p^i and t^i={t_x^i,t_y^i,t_w^i,t_h^i } which represents k anchors for a mini-batch indexed by i (0≤i<k). p_i means the probability of containing an object and t^i is the box position of anchor i. The loss function of RPN on each image is:

$$L_{RPN} = \frac{1}{N_{batch}}\sum_i L_{log}(p^i, u^i) + \lambda\frac{1}{N_{reg}}\sum_i u^i smooth_{L1}(t^i, v^i) \tag{7}$$

u^i is the ground-truth probability (1 or 0) of containing an object and v^i is the ground truth BBox of the current anchor. [18] also introduced the Intersection over Union (IoU). Suppose the IoU ratio between the predicted box and the ground truth box is less than 0.3, u^i will be set to 0. Otherwise, $u^i = 1$. In other words, the ground truth probability is decided by the overlap between the predicted and ground truth boxes. Binary entropy loss is used for calculating the loss on the classifier of RPN, and smooth L1 penalty (Eq 4.) is again used on the BBox regressor. The u^i before smooth L1 means do not consider the BBox prediction when IoU is smaller than 0.3. By default, $\lambda = 10$ and is proven to be not sensitive [18]. N_{batch} is the mini-batch size while N_{reg} is the number of valid BBox predictions (IoU ≥ 0.3).

3. Mask Dilation

Since from the result of MaskRCNN, the coverage of the mask on the target objects could not be 100%, i.e., there are possibilities that the mask does not cover some parts of the target objects. The samples are shown in Fig. 8. This may lead to lousy inpainting results, as proven by the experiments below. Dilation [30] on masks (binary images) is proposed to increase the probability of covering the whole target objects. Mask dilation expands the mask region along the border and will not harm inpainting by enlarging

DOI: 10.54254/2755-2721/2/20220668

the mask too much. Besides, dilation is not likely to significantly impact the effect of inpainting because the inpainting network LaMa [13] supports large-area masks (25%).

1. 

2. 

3. 

4. 

DOI: 10.54254/2755-2721/2/20220668

5. 

6. 

Figure 8. Sample outputs of MaskRCNN.

3.1. Large Mask Inpainting

Suvorov [13] proposed an inpainting framework named large mask inpainting (LaMa) aimed to solve the large-scale mask and high-resolution image inpainting dilemma by taking advantage of fast Fourier convolution (FFC) [31] which stands as a feed-forward network. This is a non-coarse-to-fine GAN-based method, the network architecture of LaMa can be seen in Fig. 9. The masked image (the region of the image to be inpainted is wiped) is stacked with the mask and fed into the inpainting network, and the inpainted image would be output. Each image will only go through downscale and upscale once, and the FFC modules are set into the residual block to do inpainting tasks.

DOI: 10.54254/2755-2721/2/20220668

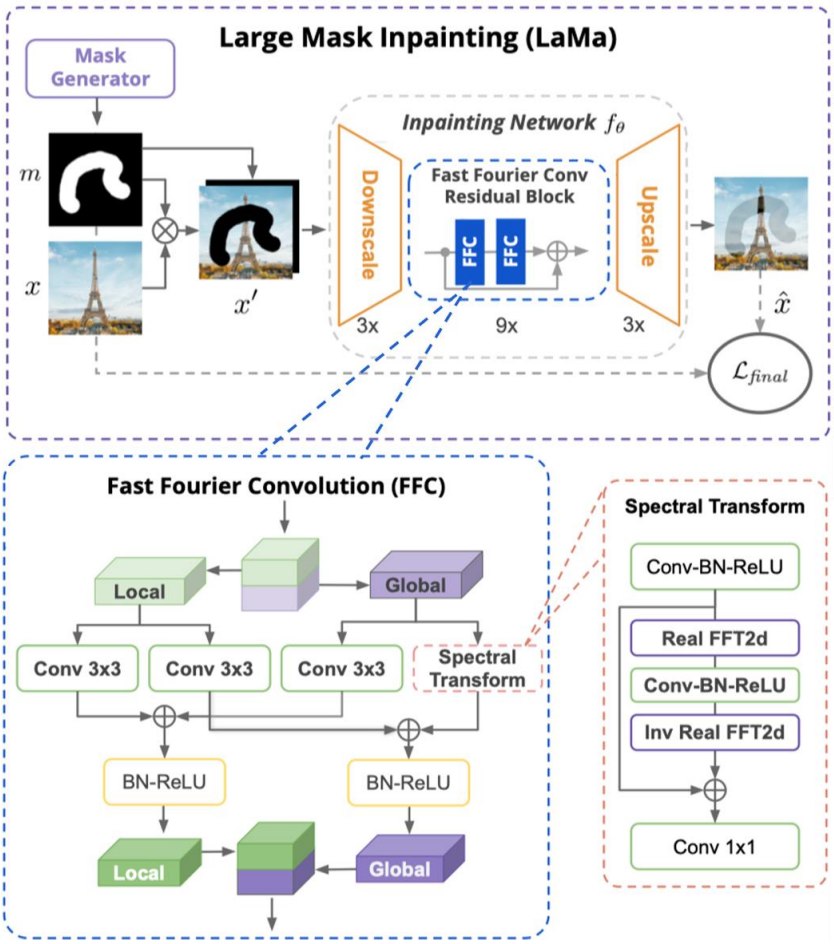


Figure 9. LaMa with FFC [13].

3.2. Fast Fourier Convolution (FFC)

The global context is essential for image inpainting, and the receptive field should be considered as broad and early as possible in the architecture, according to [13]. Thus, fast Fourier transform (FFT) [31] is brought in to contribute to the fast Fourier convolution (FFC) channel-wise. FFC has two sibling branches. One is for local context traditional convolutions, and the other is for global context using the real FFT, which only adopts half the spectrum. FFC takes a tensor produced by the input image and its mask as input, applies real FFT to it, merges the remained and inpainted parts, uses a convolution block in the frequency domain, recover a spatial structure by adopting inverse transform, and fuse the out-puts of the local and global branches. FFC achieves high efficiency and is helpful on high-resolution images [13].

3.3. Loss Functions

Since there is no ground truth for an inpainted image, the same image with the same mask could lead to different inpainting results. Other measurement methods should be adopted and contribute to the loss function to achieve better back-propagation effects. The overall weighted-sum loss function of LaMa [13] is defined as:

$$L = \alpha L_{adv} + \beta L_{HRF} + \gamma L_{Disc} + \delta R_1 \tag{8}$$

DOI: 10.54254/2755-2721/2/20220668

Adversarial Loss. Following [32], the discriminator of LaMa [13] works on the local patches broad by distinguishing "real" and "fake" patches within the inpainted image. If the patch intersects with the masked region, it will be set to "fake" by the discriminator. The adversarial loss is a non-saturating joint loss:

$$L_D(x,\hat{x},m) = -\mathbb{E}_x\left[\log D_\xi(x)\right] - \mathbb{E}_{x,m}\left[logD_\xi(\hat{x}) \odot m\right] - \mathbb{E}_{x,m}\left[\log\left(1 - logD_\xi(\hat{x})\right) \odot (1-m)\right] \quad (9)$$

$$L_G(x,\hat{x},m) = -\mathbb{E}_{x,m}\left[logD_\xi(\hat{x})\right] \quad (10)$$

$$L_{adv} = sg_\xi(L_D) + sg_\theta(L_G) \to \min_{\xi,\theta} \quad (11)$$

Among Eq 9-11., x is the input image, m is the corresponding mask, $\hat{x}$ is the inpainted image produced by the generator, $D_\xi(\cdot)$ represents the discriminator, and sg_i will stop gradients based on i.

HRF Perceptual Loss. Understanding the global context in a fast way is vital for LaMa [13]. Thus, high receptive field (HRF) perceptual loss is introduced to **achieve** this. A base model $\emptyset_{HRF}(\cdot)$ with HRF, which could be implemented by Dilated or Fourier convolutions, is introduced to be a part of the loss function:

$$L_{HRF}(x,\hat{x}) = M([\emptyset_{HRF}(x) - \emptyset_{HRF}(\hat{x})]^2) \quad (12)$$

$M(\cdot)$ computes the inter-layer mean based on the means of each layer, and $[\cdot]^2$in this case is the operation element-wise.

Other Losses. A gradient penalty loss $R1$ [33] and a perceptual loss for feature matching analysis L_{Disc} [34] are both contribute to the final loss function.

4. Methodology

A part of the test2017 of COCO [20] will be used in this experiment. Before being fed into the inpainting network, the mask will be generated by MaskRCNN and goes through a mask dilation or not. The mask is developed based on the human body, i.e., the human body region in images will be signed on the mask and painted. Since there is no ground truth for the inpainting result, the effect of the inpainting network should be manually recognized. In the experiment, the same images will be mainly processed in three ways: 1. Images are fed into MaskRCNN, and LaMa produces the results from the outputs of MaskRCNN and the images; 2. Before putting the result of MaskRCNN into LaMa, a mask dilation will process the masks. The adaptation details of MaskRCNN and LaMa will be briefly described below, and the experimental results will be displayed and analyzed in later sections.

4.1. System Adaptation

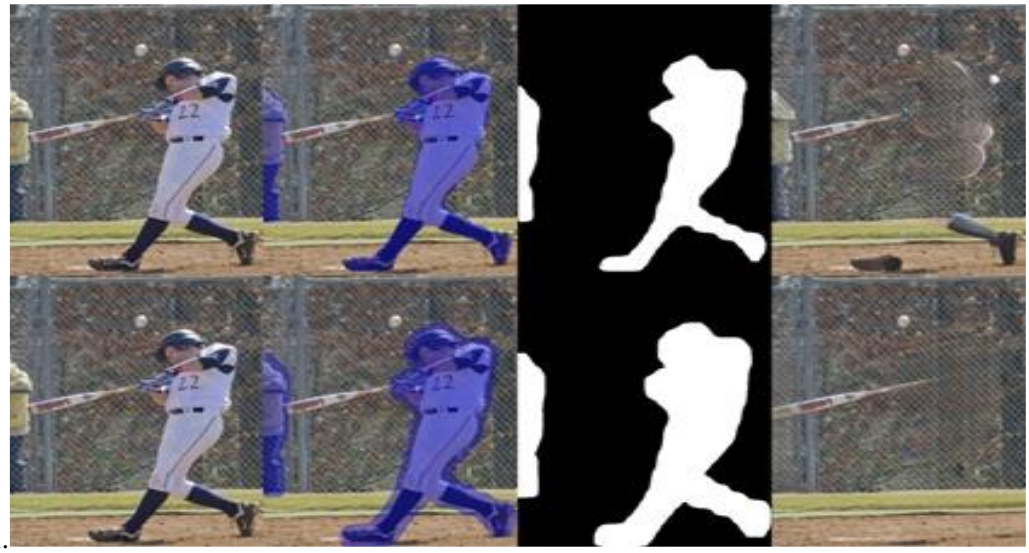
We use the pre-trained models from MaskRCNN and LaMa sources. The best result model is picked up. For MaskRCNN, the FPN-ResNet101 backbone pre-trained model is adapted for human segmentation tasks which are trained and validated by COCO [20] train2017 and val2017. For LaMa, places [24] are used as the training and validation set, making the model available for scene-based inpainting tasks.

4.2. Inference

Since this paper focuses on human removal, and the pre-trained MaskRCNN model supports multi-objects, the non-human result will be ignored, as well as the BBox result and score. The mask is the only need for further processing. Moreover, MaskRCNN achieves instance segmentation, which means a corresponding mask will be output for each instance detected, but LaMa will only take one mask for inpainting. Combining the human masks is needed. The combination of mask, $mask_{final} = mask_1 | mask_2 | \ldots | mask_n$ where n is the number of persons detected. The dilation kernel for the group with mask dilations is (20, 20), and the processing time is set to 2. The images, along with its mask, will be fed into LaMa to do image inpainting. The inference work is done on an Ubuntu 18.04 server with 8-Core Intel Xeon Platinum 8255C, Nvidia Tesla T4, and 32 GB RAM.

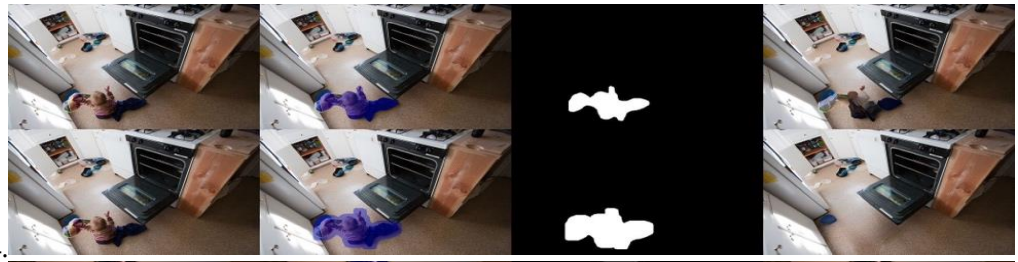
DOI: 10.54254/2755-2721/2/20220668

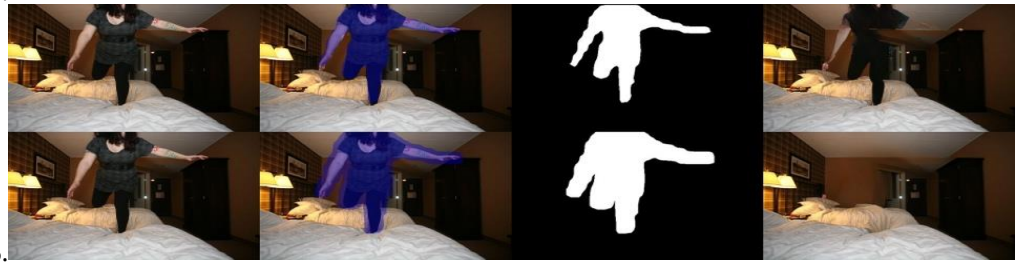
5. Results

1.


2.


3.


4.


5.


DOI: 10.54254/2755-2721/2/20220668

Figure 10. The image, masked image, mask, and inpaint result with (top) and without mask dilation (bottom).

The results in Fig. 10 show that MackRCNN could locate the person instances and draw a mask on them even if the instances do not appear entirely in the picture. However, the mask cannot cover every pixel on the target instances, which may cause low-quality inpainting results. This can be seen in the top-right image on every sub-figure in Fig. 10. Luckily, the mask dilation could address such a problem by expanding the mask, which improves the coverage. Even though there are wasted pixels (not part of the target instances but covered by the mask), the inpainting result turns out to be better: the target instances are erased, and the inpainted result seems natural (could be seen in the bottom-right image at every sub-figure.). The detailed analysis will be conducted in the sub-figure wise:

1. The original mask produced by MaskRCNN does not totally cover the player's shoes. Still, it does not cover the whole clothes of the player and the person on the player's left, which lead to LaMa considering such residue pixels and inpaint strange shapes and colors.

DOI: 10.54254/2755-2721/2/20220668

2. The similar case appears in 2. at Fig. 10, i.e., for instance segmentation without dilation post-process, the person on the surfboard is not totally covered and turned into a thinner shadow at the inpainted result. Mask dilation again solved this problem. Even though dilation caused a relatively massive waste of pixels, LaMa recovered the whole masked region and made the inpainted result indistinguishable from real images.
3. Mask dilation helped the mask to cover the whole person instances. However, the shadow of the standing person keeps here, and since most parts of the drawers on the wall are obscured by the upright person, LaMa does not do a great job of recovering them.
4. The image is well recovered. The child, his blanket, and something in his hands are regarded as the human body and masked. LaMa successfully recovers the area and looks exactly like the real image. Unlike the tennis racket reserved in 1. at Fig. 10, this could be called a lucky case.
5. The woman is mainly successfully removed, but she still has a portion of her right hand left in the inpainted image. The left side of the door frame is not appropriately recovered because the woman totally obscures it.
6. The old couple and the woman behind them are detected and masked. Even though most parts of the pavement and the fork were removed, LaMa surprisingly recovered them.
7. Both women were erased without any trace, the boards were lengthened well, and the background was naturally filled in, but the giraffe was partially erased due to occlusion, and the mask dilation caused the deer to be left unfilled.
8. Inpainted images are basically devoid of the person. Switches, door frames, and doors are effectively restored, even though the lower-left corner of the door frames is too light, perhaps due to too much shading by the person.

More results are shown in Fig. 12. which is at the end of the paper.

6. Discussion

In most cases, MaskRCNN could locate and identify the human body well, and LaMa can also repair the mask part well, according to the unmasked portion of the picture. From the comparison of the results, it is evident that the mask produced by MaskRCNN cannot be directly used on LaMa since the accuracy of the mask is not enough to be inpainted. Parts left will affect the inpainting quality. Inpainting's neural network has a low tolerance for these errors because the retained portion of the mask edge has the highest reference weight [10], so the residual body portion can significantly influence the restoration. The appearance of mask dilation can effectively alleviate this problem. Dilation clears most of the residue. Even if some parts are wasted, LaMa could repair these wasted places naturally. However, the set of procedures still has limitations on specific cases, or it does not handle anything.

6.1. Limitations

MaskRCNN Sometimes, MaskRCNN may wrongly recognize human instances. In 6. in Fig. 11, it mistakenly regards the cat as a person; in 4. and 3. in Fig. 1, it does not recognize all the human instances. And as mentioned above, the mask quality should be improved.

Mask dilation As mask dilation uses a single parameter, that is, all masks are processed twice by the (20, 20) kernel, which may lead to excessive data loss in some cases. In 3. at Fig. 11, the mask cut off the body of the ship on the right side, resulting in half of the boat being retained in the inpainted image. Such dilation parameters are too large for the smaller people and objects in the image. In this case, it is difficult for the inpainting network to fill out the incomplete objects. And the same thing can happen to any small person or object.

LaMa For angular objects, LaMa completion, can be problematic, such as 1. in Fig. 11. Because some of the changes of the lecture table were completely covered or erased, LaMa could not grasp the position of the boundary well, so the color of the lecture table extended out. Also, as mentioned above, LaMa could not handle the shadow caused by the human instances, which may affect the naturalness of inpainted images.

DOI: 10.54254/2755-2721/2/20220668

Others For images (5. and 6. in Fig. 11) with large masks, even the human brain can hardly imagine the restored content, so it is difficult to fix such images with the current technology. It is also difficult to repair the objects close to the human instance. In 4. in Fig. 11, although the bus was repaired as much as possible, the overall picture was blurred. For 2. in Fig. 11, it is hard for a human brain to imagine the specific vehicle.

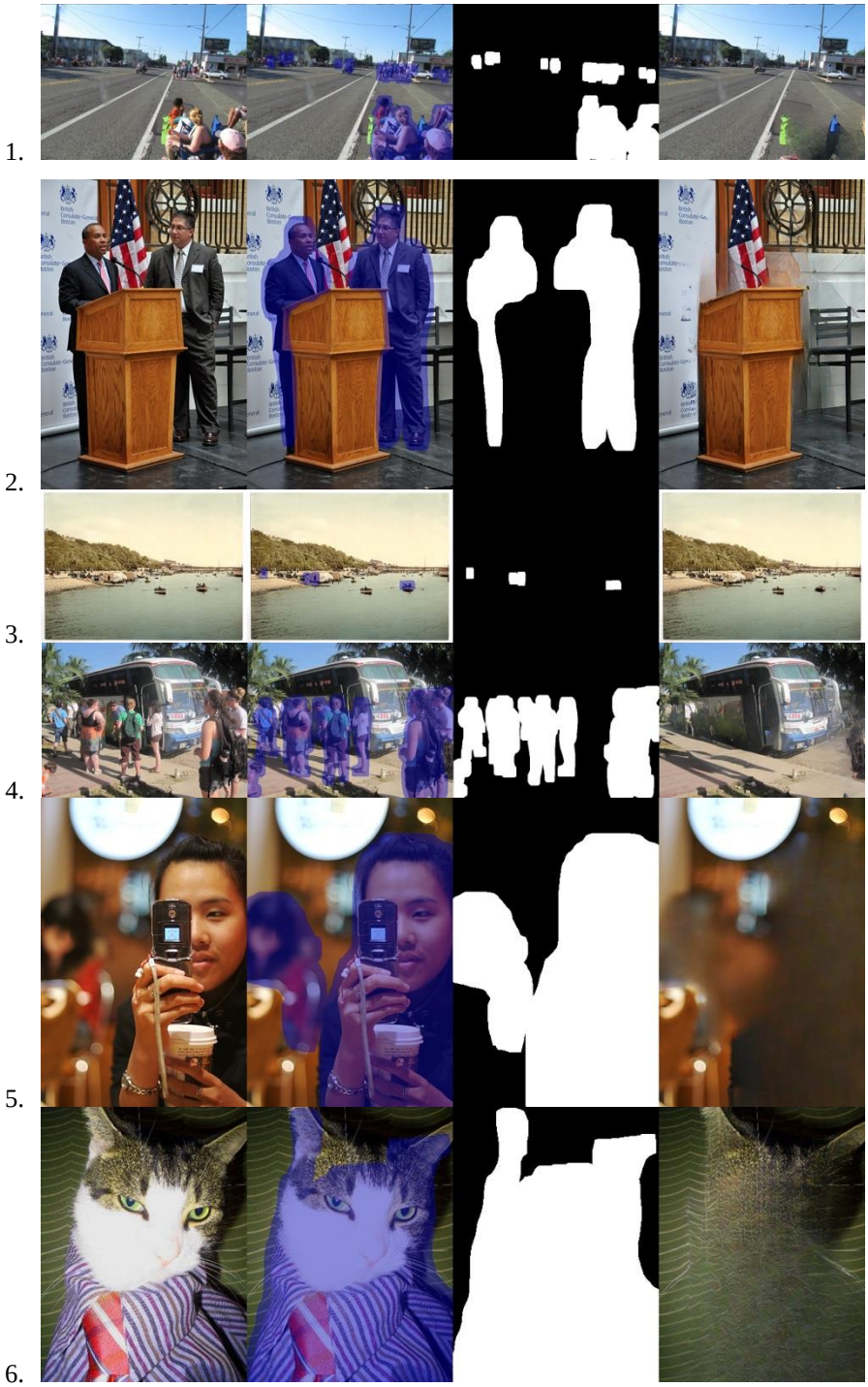
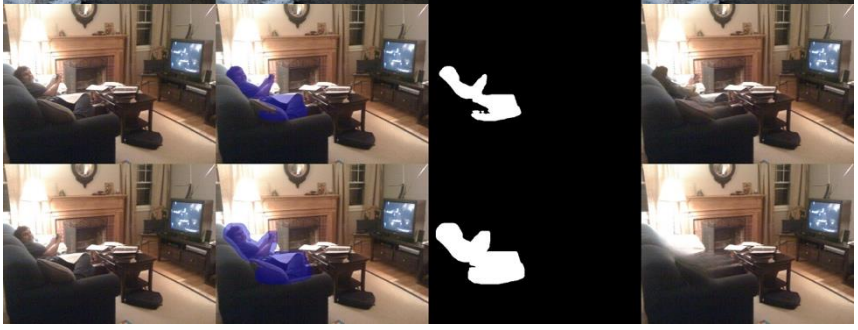


Figure 11. Bad results.

DOI: 10.54254/2755-2721/2/20220668

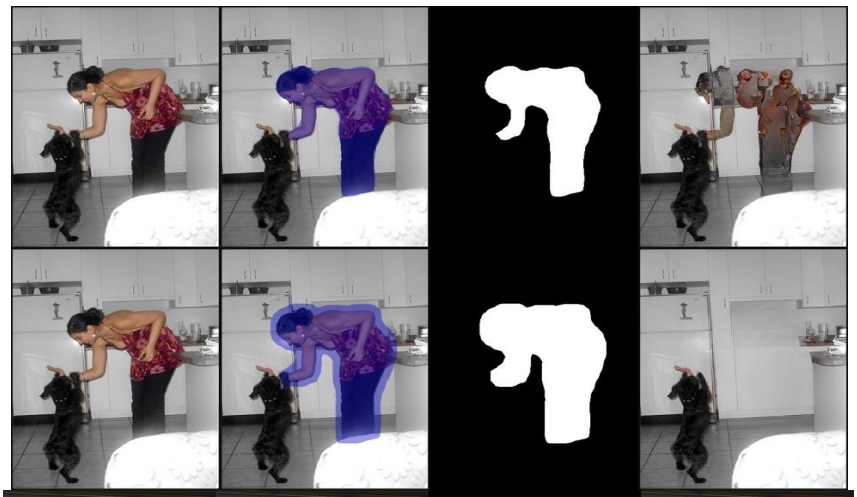
6.2. Feature Works

MaskRCNN can adopt more extensive data, more complex data and processing methods, and a longer training time to increase the accuracy of instance recognition and reduce the probability of misjudgment. Edge map can be added as a reference factor in mask drawing to increase the coverage of the mask further. For mask dilation, dynamic parameters could be adjusted by calculating the area of the mask. For example, a larger kernel and more process times are adopted for larger objects, and vice versa. The compensation effect for the mask can also be increased. For LaMa, a mechanism can be provided to ignore outliers or unexpected contents around the mask, focusing more on the global background so as not to be overly affected by residual parts.

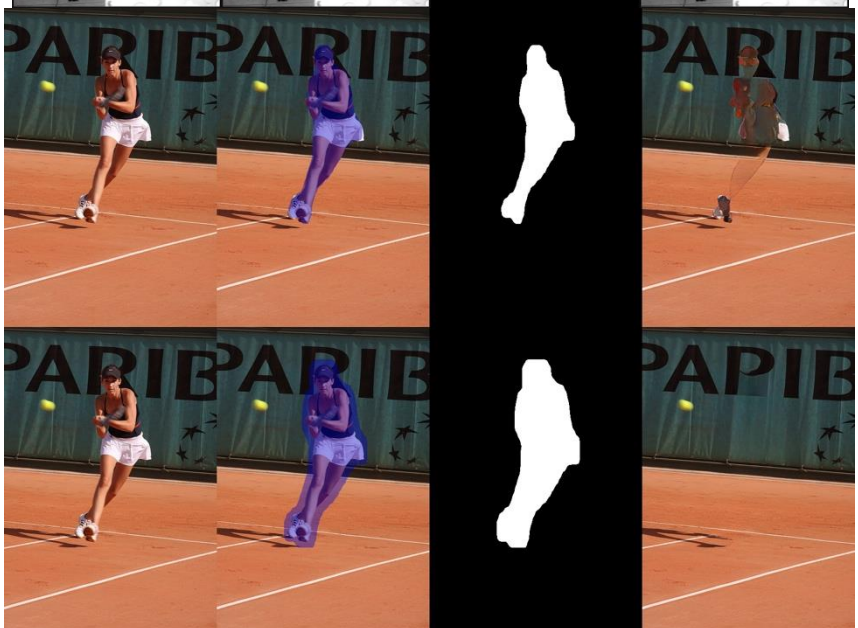
1. 
2. 
3. 
4. 

DOI: 10.54254/2755-2721/2/20220668

5.

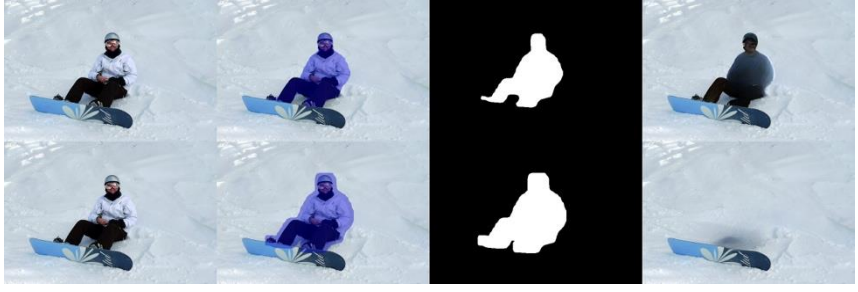


6.



7.

8.



DOI: 10.54254/2755-2721/2/20220668

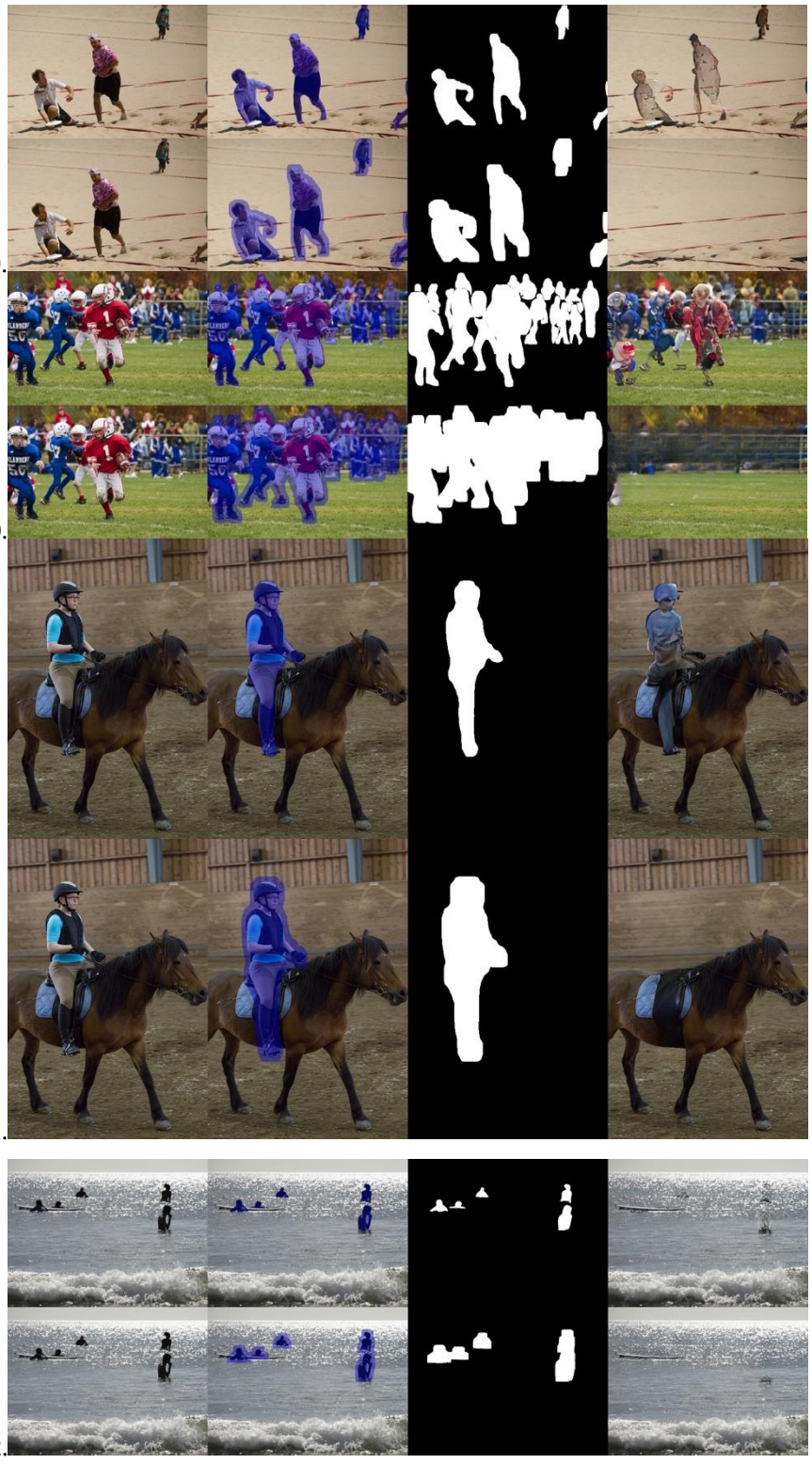


Figure 12. More results.

DOI: 10.54254/2755-2721/2/20220668

7. Result

We propose the MaskRCNN to mask dilation to LaMa method to achieve automatic human removal on images. This method is effective on COCO [20] test2017 set, even though it has a few limitations and brings many directions to make improvements. The technique could be adopted in serval areas, such as privacy protection, image restoration, photo enhancement, etc. In the future, image instance segmentation and inpainting will be upgraded, and maybe someday, mask dilation is not needed since image instance segmentation techs would draw the mask perfectly. Also, the inpainting will achieve a better effect and even can clear the shadows.

References

[1] Hafiz, A. M., & Bhat, G. M. (2020). A survey on instance segmentation: state of the art. International journal of multimedia information retrieval, 9(3), 171-189.

[2] Hariharan, B., Arbeláez, P., Girshick, R., & Malik, J. (2014, September). Simultaneous detection and segmentation. In European conference on computer vision (pp. 297-312). Springer, Cham.

[3] Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2117-2125).

[4] Bolya, D., Zhou, C., Xiao, F., & Lee, Y. J. (2019). Yolact: Real-time instance segmentation. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 9157-9166).

[5] He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. In Proceedings of the IEEE international conference on computer vision (pp. 2961-2969).

[6] Weber, M., Wang, H., Qiao, S., Xie, J., Collins, M. D., Zhu, Y., ... & Chen, L. C. (2021). Deeplab2: A tensorflow library for deep labeling. arXiv preprint arXiv:2106.09748.

[7] Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X., & Pietikäinen, M. (2020). Deep learning for generic object detection: A survey. International journal of computer vision, 128(2), 261-318.

[8] Bertalmio, M., Sapiro, G., Caselles, V., & Ballester, C. (2000, July). Image inpainting. In Proceedings of the 27th annual conference on Computer graphics and interactive techniques(pp. 417-424).

[9] Elharrouss, O., Almaadeed, N., Al-Maadeed, S., & Akbari, Y. (2020). Image inpainting: A review. Neural Processing Letters, 51(2), 2007-2028.

[10] Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., & Huang, T. S. (2018). Generative image inpainting with contextual attention. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 5505-5514).

[11] Nazeri, K., Ng, E., Joseph, T., Qureshi, F. Z., & Ebrahimi, M. (2019). Edgeconnect: Generative image inpainting with adversarial edge learning. arXiv preprint arXiv:1901.00212.

[12] Yi, Z., Tang, Q., Azizi, S., Jang, D., & Xu, Z. (2020). Contextual residual aggregation for ultra high-resolution image inpainting. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 7508-7517).

[13] Suvorov, R., Logacheva, E., Mashikhin, A., Remizova, A., Ashukha, A., Silvestrov, A., ... & Lempitsky, V. (2022). Resolution-robust large mask inpainting with fourier convolutions. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (pp. 2149-2159).

[14] Wang, K., Gou, C., Duan, Y., Lin, Y., Zheng, X., & Wang, F. Y. (2017). Generative adversarial networks: introduction and outlook. IEEE/CAA Journal of Automatica Sinica, 4(4), 588-598.

[15] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 580-587).

[16] Van de Sande, K. E., Uijlings, J. R., Gevers, T., & Smeulders, A. W. (2011, November). Segmentation as selective search for object recognition. In 2011 international conference on computer vision (pp. 1879-1886). IEEE.

DOI: 10.54254/2755-2721/2/20220668

[17] Girshick, R. (2015). Fast r-cnn. In Proceedings of the IEEE international conference on computer vision (pp. 1440-1448).
[18] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, 28.
[19] He, K., Zhang, X., Ren, S., & Sun, J. (2015). Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE transactions on pattern analysis and machine intelligence, 37(9), 1904-1916.
[20] Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Zitnick, C. L. (2014, September). Microsoft coco: Common objects in context. In European conference on computer vision (pp. 740-755). Springer, Cham.
[21] Liu, S., Qi, L., Qin, H., Shi, J., & Jia, J. (2018). Path aggregation network for instance segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 8759-8768).
[22] Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3431-3440).
[23] Canny, J. (1986). A computational approach to edge detection. IEEE Transactions on pattern analysis and machine intelligence, (6), 679-698.
[24] Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., & Torralba, A. (2017). Places: A 10 million image database for scene recognition. IEEE transactions on pattern analysis and machine intelligence, 40(6), 1452-1464.
[25] Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., & Oliva, A. (2014). Learning deep features for scene recognition using places database. Advances in neural information processing systems, 27.
[26] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition (pp. 248-255). Ieee.
[27] Xiao, J., Hays, J., Ehinger, K. A., Oliva, A., & Torralba, A. (2010, June). Sun database: Large-scale scene recognition from abbey to zoo. In 2010 IEEE computer society conference on computer vision and pattern recognition (pp. 3485-3492). IEEE.
[28] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).
[29] Jaderberg, M., Simonyan, K., & Zisserman, A. (2015). Spatial transformer networks. Advances in neural information processing systems, 28.
[30] Jankowski, M. (2006, June). Erosion, dilation and related operators. In 8th International Mathematica Symposium (pp. 1-10).
[31] Chi, L., Jiang, B., & Mu, Y. (2020). Fast fourier convolution. Advances in Neural Information Processing Systems, 33, 4479-4488.
[32] Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1125-1134).
[33] Mescheder, L., Geiger, A., & Nowozin, S. (2018, July). Which training methods for GANs do actually converge?. In International conference on machine learning (pp. 3481-3490). PMLR.
[34] Wang, T. C., Liu, M. Y., Zhu, J. Y., Tao, A., Kautz, J., & Catanzaro, B. (2018). High-resolution image synthesis and semantic manipulation with conditional gans. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 8798-8807).