

Removal of Background People from Crowded Scenery Image Using Target Detection and Refilling

Jiha Jang

Dept. of Electrical and Computer Engineering
Seoul National University
jeeit17@snu.ac.kr

Jihyung Ko

Dept. of Computer Science and Engineering
Seoul National University
hanrista1157@snu.ac.kr

Changhwi Park

Dept. of Geography
Seoul National University
smsychjy@snu.ac.kr

Junyul Ryu

Dept. of Computer Science and Engineering
Seoul National University
gajagajago@snu.ac.kr

Abstract

We present an application of human detection and inpainting to automatically remove unwanted background person(s) in photographs of crowded scenery images. Furthermore, we applied the technique with consideration of optical flow to videos. We combined object detection, face landmarks detection, and bounding box calculation to accurately determine humans from background, and distinguish the exact target person from removal target(person(s) who has to be removed) without any user intervention. The pipeline then proceeds to realistically refill the removed masked regions with neighborhood regions. As a result, the original crowded image/video is transformed to an alone photograph of the target person.

1. Introduction

The primary motivation behind this project is to provide realistic, as well as fast application to remove unwanted background person(s) from photographs. Common approaches to this problem generally include selection of each removal target and drawing region masks manually, and choosing neighborhood regions to cover the masked region. This task becomes extremely cumbersome if the photograph contains more than a few background persons or it was taken at a very crowded location such as landmarks or tourist destinations. Thus, our aim is to (1) provide an automatic process of distinguishing the ‘main’ photograph target person from other removal targets, (2) clean removal of the removal targets, and (3) construct realistic refilling of the masked regions.

2. Methodology

2.1. Human detection

The process first starts with detecting persons and distinguishing one from another. To enhance the refilling performance, we masked background persons with their contours, not their bounding boxes. By doing so, marginal pixels around masks can be preserved and later be used for repainting. Fig(1) shows what kind of result each classification or segmentation method yields. To find and distinguish persons and draw their contours, we had to perform instance segmentation to our images.

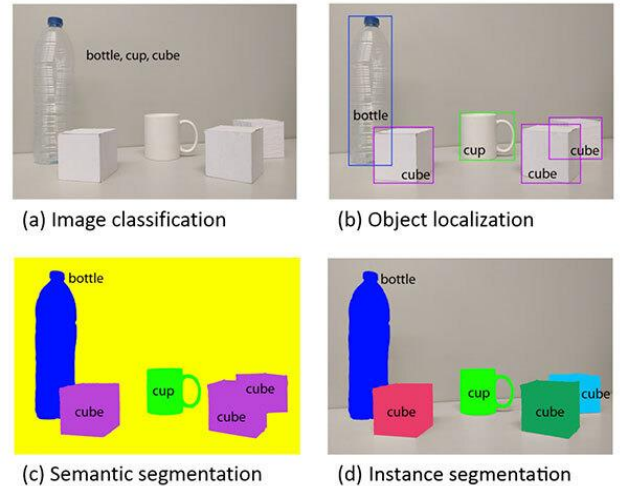


Figure 1. Example of segmentation.

Mask R-CNN is a popular instance segmentation and it is robust to noise. Our implementation code that performs human detection and segmentation is based on [8]. By modifying mask R-CNN to detect and segment only humans and tuning its hyperparameters that determines how large the segmentation will be, we can get segmentation images to pass to the next stage. Thus, the output from this step is given as in Fig(2).



Figure 2. Example of human detection

2.2. Target decision

We combined face detection and bounding box calculation to accurately distinguish the main target of the photograph from removal targets. The decision tree is summarized in Fig(3).

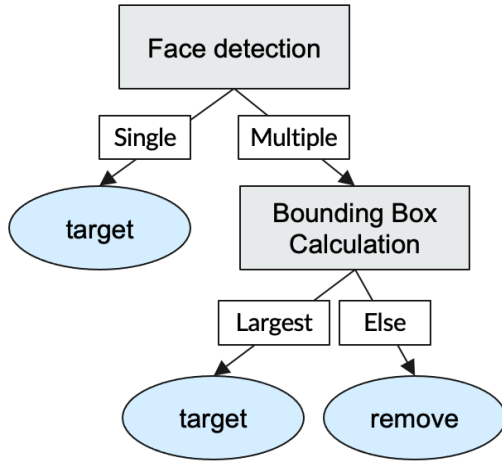


Figure 3. Target decision tree diagram

2.2.1. Facial landmarks detection

Facial landmarks detection is the baseline for distinguishing the photograph target from multiple candidates returned from human detection. We utilized dlib's face detector [9] to identify all possible facial landmarks from the image. If only single facial landmarks are detected as in Fig(4), this implies that the single detected person certainly is the main target. Thus, the process exits here and returns the landmarks detected person labeled as 'main', others as 'removal target'. If multiple facial landmarks are detected as in Fig(5) or any facial landmarks are undetected, the algorithm proceeds to the next step for accurate judgement.



Figure 4. Example of single face detection

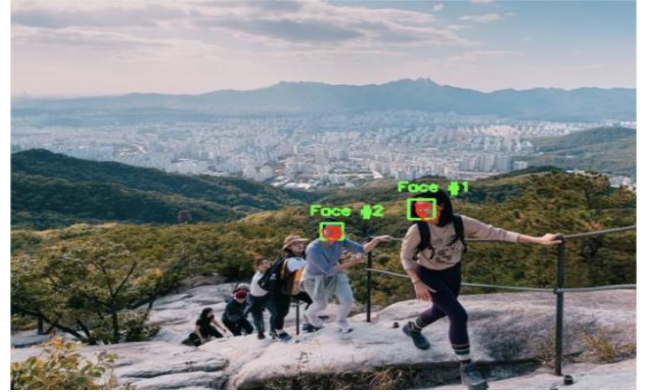


Figure 5. Example of multiple face detection

2.2.2. Bounding box calculation

Bounding box calculation is applied in order to distinguish the person with largest contour size. We applied the heuristic that it is probable for the person at the front most to be the photograph target if multiple persons are in the photo. For example, face detection result in Fig(5) requires comparison between the two persons. As index 0 person in Fig(6) possesses relatively larger contour, she is chosen as the final main target of the figure. Other contours are then regarded as removal targets and cleared out with mask.



Figure 6. Example of bounding box calculation

2.3. Refilling algorithms for masked image region

We approach this subject in two ways: a) classical method and b) deep learning method. For the performance metrics, PSNR and SSIM were used.

2.3.1. Performance metrics

PSNR(Peak-Signal-to-Noise-Ratio) represents a noise relative to a signal. The higher value, the better the image quality. PSNR can be calculated by the below expression.

$$PSNR = 10 \log \frac{s^2}{MSE}$$

SSIM(Structural Similarity) is an indicator of improving PSNR. The higher the value is, the more similar structural information the image has to the human vision. SSIM can be calculated by the below expression.

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

2.3.2. Classical method

Two classical approaches of image refilling were “texture synthesis” and “inpainting” techniques. “Texture synthesis” approach usually aims to refill large areas by replicating sample regions and “inpainting” techniques focus on filling linear structures such as lines and object contours.

The work in [1] combines these two classical inpainting methods using an exemplar-based method. Based on their key observation that refilling order is important, the algorithm calculates patch priority first. “Confidence term” measuring reliability of surrounding information of the pixel is used in this step. According to calculated patch priority, the algorithm finds a matching block to fill the top-priority patch and refill the patch. It finds the “closest block” with SSD. Finally, the confidence term of every single patch is updated for the next step. These steps are recursively done until all patches are filled.

We improved the classic refilling method implemented in [2] in two ways based on the work in [3], [4]. First, based on the work in [3], we improved the patch priority function by considering curvature of isophotes additionally. By doing so, we could better reflect local characteristics of the image in the patch priority. And based on the work in [4], we improved the method finding matching blocks by using both Euclidean distance and SSD.

2.3.3. Deep learning method

To refill the erased parts, deep learning method of free-form image inpainting with gated CNN was applied to this subject. This method aims at free-form image inpainting. Free-form image inpainting refers to a task that naturally

fills multiple holes with arbitrary positions and shapes. It has two classes: (1) Gated CNN trains soft mask from data automatically. and (2) SN-PatchGAN which reduces the problem of being biased toward a specific class.

For implementation, we referenced [5] implementing free-form image inpainting with gated convolution. And we used the pretrained model Places2 which is trained with images of resolution 256x256 and largest hole size 128x128, above which the results may be deteriorated.

2.4. Video inpainting

The human detection and inpainting methods applied to images are also applicable to videos. Automated inpainting human detection and masking makes it feasible to efficiently apply background person inpainting to videos.

The very first naive method that we can come up with regarding the video inpainting is a frame-by-frame inpainting, in which a video is just broken up into frames and undergoes through the above methods. Once the video is decomposed into consecutive frames composing it, each frame is regarded as a normal image. Humans are extracted from the frame and distinguished into a target and background persons and masks the latter ones. Then the masked regions are refilled with the methods above. The resulting frames are collected again to yield a new inpainted video.

The naive video repainting method does repaint videos but has two drawbacks; waving artifacts and high computational cost. Since we have regarded each frame as an image, they were processed independently, eventually neglecting temporal data which is absent in images but is inherent to videos. Even if our method is deterministic, each masked frame cannot not differ when they are inpainted and this leads to the unwanted perturbation. Furthermore, processing each frame one by one requires too much computing power, especially when it comes to the classical method, which takes over an hour to inpaint a single frame.

Now we suggest a new video inpainting to exploit the temporal data in videos. The basic idea is that frames in a video are interrelated and those relations can be presented in a matrix form. It can be assumed that changes between frames can be approximated as affine transforms and those changes are small enough. Under these assumptions, the workflow of the new method is shown below in Fig(7).

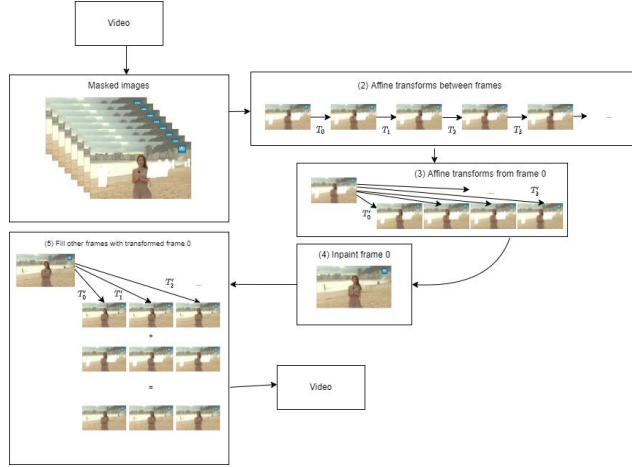


Figure 7. Video inpainting process

The new method repaints only the first frame of a video and continually reuses it but with transformations. This approach not only reduces the computation loads but also reduces any unnecessary perturbation between frames.

3. Results

3.1 Image refilling

3.1.1 Improved classical method

Table 1,2,3 show that the improved algorithm performs refilling in a better way. Our evaluation metrics, PSNR and SSIM improved and we can observe some improvements in refilling. For example, in Table 2 we can see that the original refilling algorithm failed to refill mask region shaped “A” at bottom left of the image. But our improved algorithm refilled the mask region well so we can’t find the “A” shape in the result image of the improved algorithm.

| | refilling by original algorithm | refilling by improved algorithm |
|-------|---------------------------------|---------------------------------|
| image | | |
| PSNR | 34.52 | 34.60 |
| SSIM | 0.8355 | 0.8477 |

Table 1. Original and improved refilling result Comparison

| | refilling by original algorithm | refilling by improved algorithm |
|-------|---------------------------------|---------------------------------|
| image | | |
| PSNR | 35.73 | 35.74 |
| SSIM | 0.8089 | 0.8104 |

Table 2. Original and improved refilling result Comparison

| | refilling by original algorithm | refilling by improved algorithm |
|-------|---------------------------------|---------------------------------|
| image | | |
| PSNR | 41.29 | 41.32 |
| SSIM | 0.9508 | 0.9543 |

Table 3. Original and improved refilling result Comparison

3.2.2 Deep learning method

Table 4 shows our deep learning image refilling results step by step.

| | image 34-0 | image 17-0 |
|-------------|------------|------------|
| original | | |
| mask erased | | |

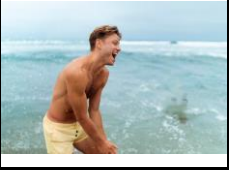

| | image 34-0 | image 17-0 |
|-----------|---|---|
| repainted |  |  |
| PSNR | 41.75 | 41.42 |
| SSIM | 0.9678 | 0.9574 |

Table 4. The results of repainting about two dataset images and metric values

3.2.3 Image refilling results comparison

Table 5 shows results of three different image refilling methods when applied to our dataset. We can find stair shaped artifacts in results of classical refilling method. But a result of improved classic refilling method shows more smooth refilling regarding the stair shaped artifacts. Classical refilling method took more time than deep learning method due to matrix calculations.



Table 5. Refilling results comparison

3.2 Video refilling

The new video inpainting method successfully reduces consequent artifacts that was unintentionally produced when optical flow was not considered. Still not all the perturbations can be ultimately eliminated but at least the refilling looks consistent. Also, the total time consumed to inpaint and reproduce the video has been significantly reduced. Thus, the final result has improved both in time and quality basis.

4. Discussion

To erase the removal target person(s) from image or video automatically, we applied several methods. We detected human objects and selected a target person to remain by facial recognition and bounding box size. After removing the bounding box(es) of removal person(s), we refilled the region by classical method and deep learning method. Our deep learning method was faster and more effective (about 5db higher of PSNR) than the classical method in our datasets. But deep learning method has the problem that we cannot infer the reason why some faults occur. So, Further, we will improve some classical methods. Completing the case of the image, we tried to apply it on video. But there was a problem that video got nauseated as refilling was applied independently for each frame. To solve this problem, we use optical flow which is learned at our lecture. As a result, we could obtain a more natural result.

In the future, we can improve in four ways. First, we can improve our target detection algorithm by optimizing segmentation and face detection methods. We found some examples that target detection was failed due to limitations of segmentation and face detection. For example, the face detection method we used couldn't find target when the target is wearing sunglasses or when the lateral face is shown in the picture. Also, the segmentation method sometimes made two bounding boxes for one target, obscuring our target finding process. Comparing existing methods and deep learning methods will help find better performance. Also, in our work, target detection was based on heuristic method. But we can also develop target detection by teaching deep learning model tendencies or characteristics of the main target in pictures. Second, in the classical refilling method, we can improve the algorithm by adjusting patch size at each refilling step or further optimizing matching block finding method. Third, in the refilling method by deep learning, we can train new models with our datasets to suit our case more properly. Finally, in video refilling, we can come up with additional ways to get rid of artifacts.

References

- [1] Criminisi, Antonio, Patrick Pérez, and Kentaro Toyama. "Region filling and object removal by exemplar-based image inpainting," IEEE Transactions on image processing 13.9, pp. 1200-1212, 2004.
- [2] Igorcmoura, mgw6. Inpaint-object-remover. <https://github.com/igorcmoura/inpaint-object-remover>, 2021.
- [3] Yu-Ting He, Xiang-Hong TANG. "Color Image Inpainting By an Improved Criminisi Algorithm." ITM Web of Conferences, 12(05023), 2017.
- [4] Song Yuheng, YAN Hao. "Image Inpainting Based on a Novel Criminisi Algorithm." arXiv:1808.04121, 2018.

- [5] JiahuiYu. Generative Image Inpainting.
https://github.com/JiahuiYu/generative_inpainting
- [6] Baker, S. and Matthews, I, "Lucas-kanade 20 years on: A unifying framework: Part 1," Technical Report CMU-RI-TR-02-16, Carnegie Mellon University Robotics Institute, 2002.
- [7] He, Kaiming, et al. "Mask r-cnn," Proceedings of the IEEE international conference on computer vision, 2017.
- [8] Satya, Mallick. Mask-RCNN.
<https://github.com/spmallick/learnopencv/tree/master/Mask-RCNN>, 2021.
- [9] Davis E. King. Dlib-ml: A Machine Learning Toolkit. Journal of Machine Learning Research 10, pp. 1755-1758, 2009
<https://github.com/davisking/dlib>