



IMAGE GENERATION METHOD FOR OBJECT DETECTION IN CONTOUR CLEAR SCENE

Rui Huang

School of Computer Science and Technology, Tianjin Polytechnic University, Tianjin 300387, China.

ABSTRACT

Correct detection of blurred objects has always been a difficult problem for image detection and recognition. Most of the existing methods are to identify fuzzy objects through extensive training by constructing new algorithms. In this paper, we provide a new method to generate semantic layouts by dividing the image of a blurred object according to the contour of the object, then set different labels on the occlusion object, and then generate photographic images using the CRN network. Use the classification network to effectively detect the most likely objects. In more details, our method of synthesizing image is not like traditional methods relying on adversarial training to synthesizing photographic images. Our method uses a single feedforward network, trained end-to-end with a direct regression objective, as a rendering engine, use the 2D semantic specification of the scene to generate the corresponding photographic image. The experimental results on the Cityscapes data set shows that our method can generate the high quality images that is as good as the state of art method while it solves the problem of the awareness of the objects.

KEYWORDS: CRN network, Mask R-CNN, Mask Scoring R-CNN, Photographic images.

INTRODUCTION:

For image detection and recognition, detecting blurred objects in a picture that only see the outline is a very challenging technical problem. We first reference segmentation and generate a method for retesting. The difficulty of directly detecting objects can be avoided very well. A cascaded optimization network for our generation method. In the generation photographic images of real images, commonly used are unsupervised models, such as GAN, VAE [17,18,20] and so on. The state of art method [paper] However, for the direct use of the feed forward convolution network to train, the regression loss function is minimized, thereby realizing the conversion from the semantic label map of the image segmentation result to the original cityscapes. It trains generators and discriminator networks without relying on GAN, instead of using a Cascaded Refinement Network (CRN) to generate realistic cityscapes. The goal of the model is to map semantic layout, w(wide), h(long), c(the number of semantic categories, such as: pedestrians, cars, trees, buildings, etc.), the purpose is to generate a w(wide), h(long), RGB(three channels) color-realistic images. On the other side, our method relies on the awareness of the labels for the objects that may limit the utilization in the real applications. For example, in terms of image generation, pictures of flowers [28] and birds [29] are generated. These methods work well by generating data sets on specific tag classes. In the data set of complex images, because more tags need to be translated into pixels, the boundaries of each class are not obvious. The perceived quality of the generated is reduced.

To solve this problem, our method uses a single feedforward network, trained end-to-end with a direct regression objective, as a rendering engine, use the 2D semantic specification of the scene to generate the corresponding photographic image. The experimental results on the Cityscapes data set shows that our method can generate the high quality images that is as good as the state of art method while it solves the problem of the awareness of labels of the objects.

The method used for our object detection is Mask R-CNN. Object detection [23,24,25,26] is the combination of objects detected by computer vision tasks, classifying targets and using borders to locate each target. So object detection can correctly segment each target in the image. At the same time, all targets in the image are correctly detected. Research shows that Mask R-CNN has the advantages of simplicity, flexibility and speed, beyond the previous. The most advanced target segmentation results. The Mask R-CNN method is extended on the basis of Faster R-CNN. Compared with Faster R-CNN, Mask R-CNN adds a branch that predicts the segmentation mask on each region of Interest (RoI). It is used for parallel execution of branches for classification and target detection frame regression.

This paper is organized as follows. We briefly review the related work in the second section, and in the third section provide the framework of our model and the experimental method. Our experimental results are discussed in the fourth section.

RELATED WORK:

The most prominent method of image synthesis in contemporary times is the Generative adversarial networks (GANs) created by Goodfellow et al[1], which is used to synthesize MNIST digits and 32x32 images, aiming to reproduce the appearance of different classes of CIFAR-10 dataset. Denton et al.[2]. By using the Laplacian pyramid structure, each level uses a separate GAN model, and the model is independently trained to synthesize the details of its scale. By using this method, the pyramid of the image can be increased to 96x96, and the synthesized image is smoother. This work is an important prerequisite for us because multi-scale transformation is a core feature of the cascading refining network approach. The main difference is that the cascading refining network trains a single model end-to-end to directly synthesize the output image and does not use confrontation training. Cascaded refining networks have global coordination, high resolution, and memory. It achieves a sufficiently high resolution through a dedicated resolution multiplier module. At the same time increase network capacity to improve generalization capabilities. The loss function used by this network is as follows:

$$\sum_{p=1}^c \min_u \sum_l \lambda_l \sum_j \|L_p^l \odot (\Phi_l^j(I) - \Phi_l^j(g_u(L; \theta)))\|_1,$$

In each of the semantic classes p in the loss function, let L_p denote the corresponding channel $L(\dots, p)$ in the input label map. Thereby determining a more powerful loss of diversity. The CRN network uses tags as parameters in the loss function, so we need to detect the objects in the picture through Mask R-CNN to identify the tags in the picture.

Mask R-CNN: The Mask R-CNN method performs target detection by finding a certain number of candidate regions[3] [4] and performing convolution on each RoI independently[5]. Based on the improvement of R-CNN[6] [7], using RollPool to select Roll on the feature map achieves faster speed and better accuracy. Faster R-CNN generates candidate frames by using the RPN learning attention mechanism. There are also subsequent improvements to the flexibility and robustness of Faster R-CNN [8]. Recently, Li et al[9] [10]. merged the segmentation candidate system with the target detection system by Fully Convolutional Instance Segmentation

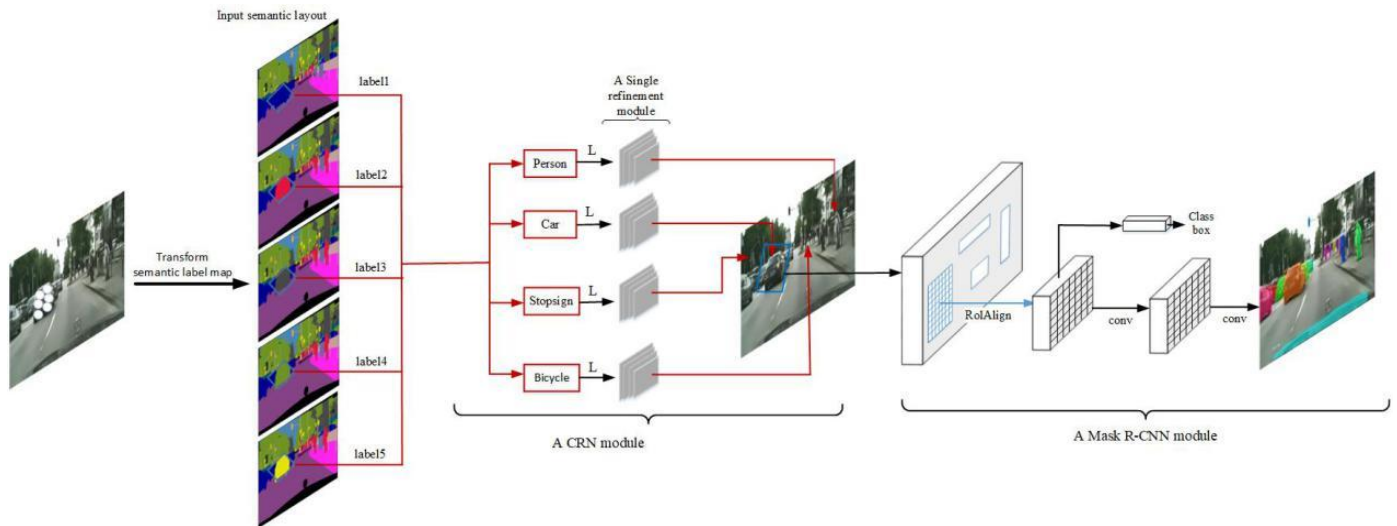
(FCIS) [11]. A set of position sensitive output channel candidates is obtained with full convolution. These channels handle both target classification, target detection and masking, which makes the system faster.

$$L_{final} = L(\{p_i\}, \{t_i\}) + (L_{cls} + L_{box} + L_{mask})$$

In the above formula, both L_{box} and L_{mask} will work for the positive RoI. In Mask R-CNN, there are some slight adjustments compared to Faster R-CNN. For example, positive RoI is defined as IoU greater than 0.5 with Ground truth (0.7 in Faster R-CNN)

OUR METHOD:

The framework:



GENERATING PHOTOGRAPHIC IMAGES:

Preliminaries:

Consider a semantic layout $L \in \{0,1\}^{m \times n \times c}$, where $m \times n$ is the pixel resolution and c is the number of semantic classes. Each pixel in L is represented by a one-hot vector that indicates its semantic label: $L(i,j) \in \{0,1\}^c$ s.t. $\sum_p L(i,j,p) = 1$. One of the c possible labels is 'void', which indicates that the semantic class of the pixel is not specified. Our goal is to train a parametric mapping g that given a semantic layout L produces a color image $I \in \mathbb{R}^{m \times n \times 3}$ that conforms to L .

In the course of this project we have experimented with a large number of network architectures. As a result of these experiments, we have identified three characteristics that are important for synthesizing photorealistic images. We review these characteristics before describing our solution.

Global coordination:

Globally consistent structure is essential for photorealism. Many objects exhibit nonlocal structural relationships, such as symmetry. For example, if the network synthesizes a red light on the left side of a car, then the corresponding light on the right should also be red. This distinguishes photorealistic image synthesis from texture synthesis, which can leverage statistical stationarity [29]. Our model is based on multi-resolution refinement. The synthesis begins at extremely low resolution (4×8 in our implementation). Feature maps are then progressively refined. Thus global structure can be coordinated at lower octaves, where even distant object parts are represented in nearby feature columns. These decisions are then refined at higher octaves. High resolution. To produce truly photorealistic results, a model must be able to synthesize high-resolution images. Low resolution is akin to myopic vision in that fine visual features are not discernable. The drive to high image and video resolutions in multiple industries is a testament to resolution's importance. Our model synthesizes images by progressive refinement, and going up an octave in resolution (e.g., from 512p to 1024p) amounts to adding a single refinement module. The entire cascade of refinement modules is trained end-to-end.

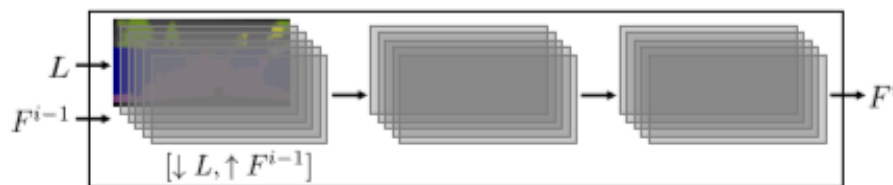


Figure 3. A single refinement module.

Memory:

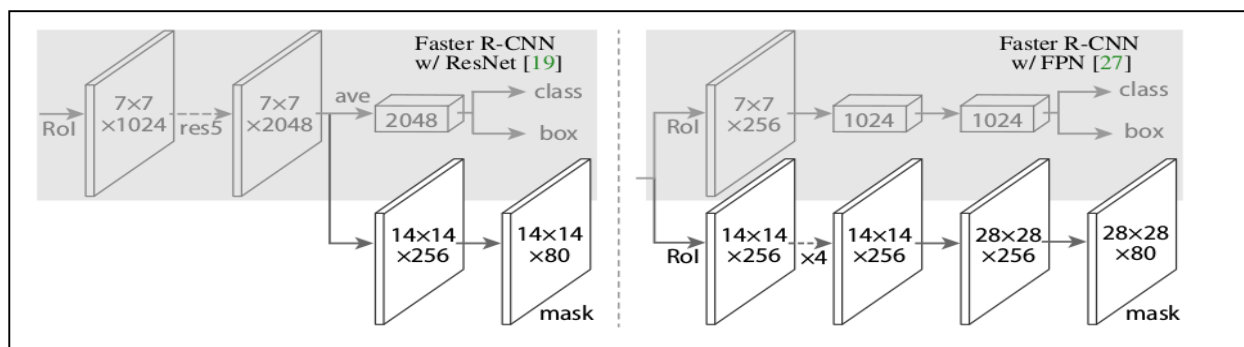
We conjecture that high model capacity is essential for synthesizing high-resolution photorealistic images. Human hyperrealistic painters use photographic references as external memory of detailed object appearance [21]. The best existing image compression techniques require millions of bits of information to represent the content of a single high-resolution image: there exists no known way to reconstruct a given photograph at high fidelity from a lower capacity representation [30]. In order for our model to be able to synthesize diverse scenes from a given domain given only semantic layouts as input, the capacity of the model must be sufficiently high to be able to reproduce the detailed photographic appearance of many objects. We expect a successful model to reproduce images in the training set extremely well (memorization) and also to apply the learned representations to novel layouts (generalization). This requires high model capacity. Our design is modular and the capacity of the model can be expanded as allowed by hardware. The network used in most of our experiments has 105M parameters and maximizes available GPU memory. We have consistently found that increasing model capacity increases image quality. 3.2. Architecture The Cascaded Refinement Network (CRN) is a cascade of refinement modules. Each module M^i operates at a given resolution. In our implementation, the resolution of the first module (M^0) is 4×8 . Resolution is doubled between consecutive modules (from M^{i-1} to M^i). Let $w_i \times h_i$ be the resolution of module i . The first module, M^0 , receives the semantic layout L as input

(downsampled to $w_0 \times h_0$) and produces a feature layer F^0 at resolution $w_0 \times h_0$ as output. All other modules M^i (for $i \neq 0$) are structurally identical: M^i receives a concatenation of the layout L (down sampled to $w_i \times h_i$) and the feature layer F^{i-1} (upsampled to $w_i \times h_i$) as input, and produces feature layer F^i as output. We denote the number of feature maps in F^i by d_i . Each module M^i consists of three feature layers: the input layer, an intermediate layer, and the output layer. This is illustrated in Figure3. The input layer has dimensionality $w_i \times h_i \times (d_{i-1} + c)$ and is a concatenation of the downsampled semantic layout L (c channels) and a bilinearly up sampled feature layer F^{i-1} (d_{i-1} channels). Note that we do not use up convolutions because up convolutions tend to introduce characteristic artifacts [30]. The intermediate layer and the output layer both have dimensionality $w_i \times h_i \times d_i$. Each layer is followed by 3×3 convolutions, layer normalization [2], and LReLU nonlinearity [23].

The output layer F^T of the final module M^T is not followed by normalization or nonlinearity. Instead, a linear projection (1×1 convolution) is applied to map F^T (dimensionality $w_T \times h_T \times d_T$) to the output color image (dimensionality $w_T \times h_T \times 3$). The total number of refinement modules in a cascade depends on the output resolution. For our main experiments on the high-resolution Cityscapes dataset, the number of modules is 9, accounting for a resolution increase from 4×8 to 1024×2048 . For the number of feature maps d_i , we use 1024 for $i = 0, 4$, 512 for $i = 5, 6$, 128 for $i = 7$, and 32 for $i = 8$.

Mask R-CNN:

The Mask R-CNN used for detection later consists of two phases. The first phase is called the Region Proposal Network (RPN), at which the candidate target bounding box is proposed. In the second phase, in parallel with the prediction class and the box offset, Mask R-CNN also outputs a binary mask for each RoI. This is in contrast to other recent systems, whose classification depends on mask prediction [12]. For the network architecture, a lower convolutional network for feature extraction on the entire image is used, as well as an upper layer network for detecting frame identification and mask prediction. The lower layer network uses ResNet-FPN for feature extraction. The upper layer network adds a full convolutional mask prediction branch to extend the upper layer network of the Faster R-CNN. Details as shown below



EXPERIMENTS:

Training:

The Cascade Refining Network (CRN) uses the semantic segmentation data set $D = \{I, L\}$ for training. The semantic layout map L is used as an input, and the corresponding color image I is taken as an output. The image feature is extracted by means of a VGG-19 image perception model, which mainly extracts certain layers of different image features recognized by it as the basis for calculating the training loss, thereby covering the low-level fine-grained features such as edges and colors in the image features. And the advanced layout features of objects, categories, etc., thus constructing a comprehensive and powerful loss function. [13]

$$\mathcal{L}_{I,L}(\theta) = \sum_l \lambda_l \|\Phi_l(I) - \Phi_l(g(L; \theta))\|_1.$$

$\{\Phi_l\}$ is a set of layers in the network Φ , for example Φ_0 represents an input image. Each layer is a three-dimensional tensor. g is trained by the image generation network, and θ is a collection of this network parameter. The hyperparameters $\{\lambda_l\}$ is to balance the contribution of each layer L to the loss. Each layer initializes a number as the initial hyperparameters $\{\lambda_l\}$. After 100 iterations, the hyperparameter is re-adjusted to normalize the contribution to the loss function.

Mask R-CNN training uses image-centric training [7]. The image is scaled (shorter side) to 80 pixels. The batch size is 2 images per GPU, each image has N RoI samples, and the ratio of conquered samples is 1:3. The N of the C4 lower layer network is 64, and the FPN is 512. We used 16 GPUs to train 160k iterations with a learning rate of 0.02, and the learning rate was divided by 10. at 120k iterations. We used a weight attenuation of 0.0001 and a momentum of 0.9.

The most reliable known method for assessing the authenticity of a composite image is the perceptual experiment of a human observer. These experiments produce quantitative results and have been used for related work. Others have tried to design automatic measurements to assess the authenticity of the loop. We have tried to apply the Mask R-CNN network to automatically evaluate the authenticity of images generated by the CRN network.

	Training data	AP[val]	AP	AP ₅₀	person	rider	car	bus	bicycle
InsuranceCut	fine+coarse	15.8	13.0	27.9	10.0	23.7	8.0	9.3	4.7
DWT	fine	19.8	15.6	30.0	15.1	32.9	11.7	7.9	4.9
SAIS	fine	-	17.4	36.7	14.6	35.7	12.9	10.3	7.8
DIN	fine+coarse	-	20.0	38.8	16.5	25.7	16.7	17.1	10.1
SGN	fine+coarse	29.2	25.0	44.9	21.8	39.4	20.1	17.7	12.4
Mask R-CNN	fine	31.5	26.2	49.9	30.5	46.9	23.7	19.1	16.0
Mask R-CNN	fine+COCO	36.4	32.0	58.1	34.8	49.1	27.0	24.1	18.7

The above table compares the existing techniques on Mask R-CNN with val and test sets. The Mask R-CNN test achieved 26.2AP, which was 30% higher than the previous DIN, and the Mask R-CNN achieved an improvement of about 50% compared to the (17.4AP) using fine data.

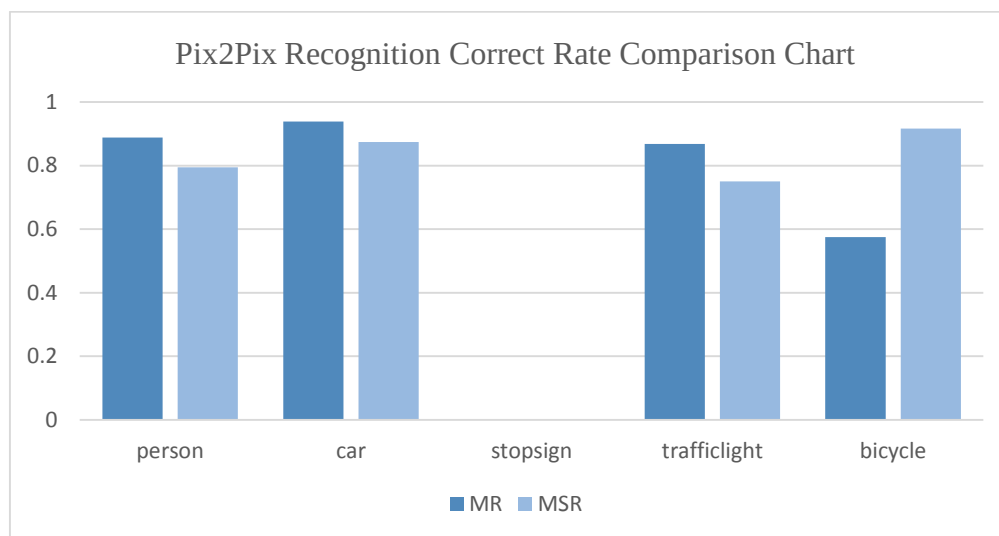
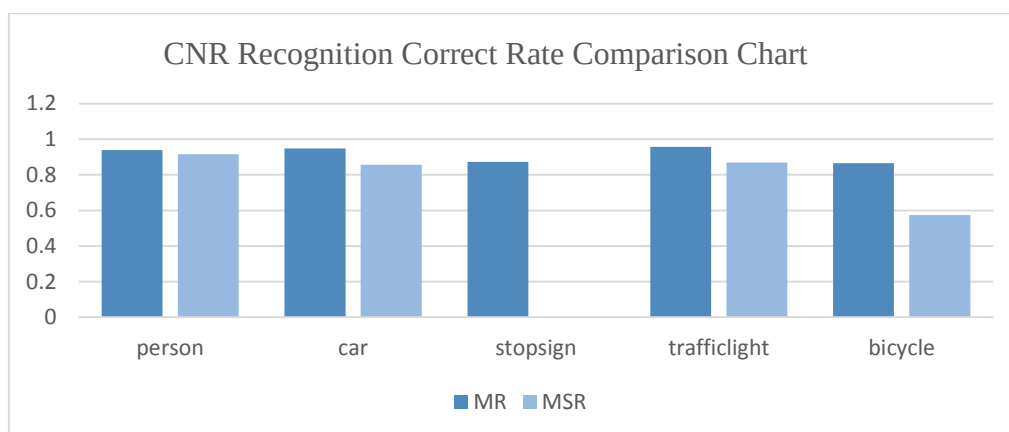
RESULTS:

We use a picture with a blurred object to generate a realistic layout image of the real image through the image segmentation network. The Mask-RCNN network detects and identifies the five labels with the highest probability of blurry objects in the real picture, and then changes the label of the blurred object in the semantic

layouts picture to the five labels, and then generates the network through the trained CRN. We prepared a photo layout image of the semantic layouts with five different labels. Finally, Mask R-CNN is used for identification and detection. Finally, the Mask R-CNN is used for identification and detection. Since we use the samples of the cityscapes dataset to train and test the CRN network, we use the Mask R-CNN network trained with the COCO dataset for detection and identification when generating image evaluations.

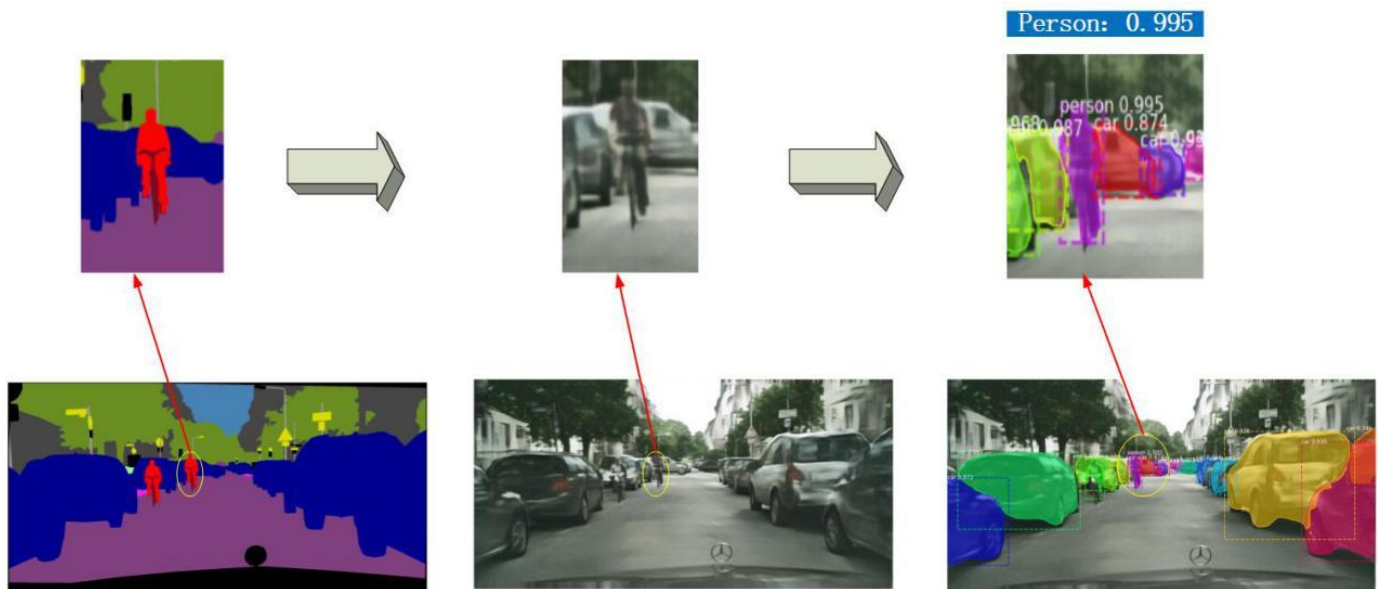
According to the results of the test by Mask R-CNN, we obtain the picture content of CRN synthesis. The detection of Mask R-CNN can achieve a realism of 0.998. Through the comparison of other interference content, we have found that the effect of CRN generating picture network is very good. For example, vehicles with different positions and colors on the same picture, the realism of the vehicle generated by the first picture is 0.998, and the second image is not generated by the interference item of the vehicle category, is detected as an elephant and the realism is 0.881. The third and fourth images that were not generated by the interference items of the vehicle category were not detected. The experimental results show that as long as a correct pixel-level segmentation image is given, the realism of the CRN-generated photo-level image can reach more than 90%.

	Test data	Person	car	stop sign	traffic light	bicycle
Mask R-CNN	CRN images	0.939	0.948	0.873	0.957	0.866
	Pix2PixHD images	0.888	0.939	0	0.908	0.90
MaskScoringR-CNN	CRN images	0.916	0.856	0	0.868	0.575
	Pix2PixHD images	0.794	0.874	0	0.75	0.916



MR in the above table is the Mask R-CNN detection method introduced in our paper, and MSR is another detection method of Mask Scoring R-CNN[26]. From the results, compared with the Mask R-CNN method. We use the same training set to train two detection methods. However, in the detection category, the MR can detect the stop sign category, and the MSR is not detected in the entire test sample. Mask scoring R-CNN works better for image depth detection, but for continuous car-to-car situations, the overlapping car parts are detected as multiple cars during the test. When Mask R-CNN detects our samples, it is better than the Mask scoring R-CNN when the vehicles on the street overlap each other. So from the results, our method has a better effect in detecting the original image than the Mask R-CNN method we use.

The whole experiment flow is that we first train the CRN network using the cityscapes data set. When we train the CRN network, we use our prepared test samples to generate photo-level images. Finally, the Mask R-CNN is used for identification and detection. Since we used the samples of the cityscapes dataset to train and test the CRN network, we used the Mask R-CNN network trained with the COCO dataset for detection and identification when generating image evaluations.



The above figure is a partial pedestrian generation and detection process amplification for more convenient viewing of the results of the generation and detection of content in the picture. It can be seen from the test result of 0.995 that the generation effect of the CRN generation network is very good. Therefore, through our segmentation generation re-detection method to identify fuzzy objects that can only see the outline, it has a good effect.



A1: layout: red object

A2: layout: green object

A3: layout: red object



B1: photographic images

B2: photographic images

B3: photographic images



C1: Object 1: person: 0.998

C2: Object 1: person: 0.863

C3: Object 1: person: 0.995

Object 2: Person: 0.989

Object 2: fire hydrant: 0.887

Object 2: person: 0.968

CONCLUSION:

By segmenting and generating re-detection methods to identify fuzzy objects that can only see the outline, our method synthesizes the original images of different labels of the same semantic-level segmentation layout image, and then detects the generated images through Mask R-CNN. Identification, by selecting the original semantic level segmentation layout image with the highest probability value of the object in the picture, and comparing whether to conform to the original picture, thereby judging the feasibility of the image synthesis network based on the pixel semantic layout. By judging this synthesis method, it is more meaningful for the application of post-exploration photographic image synthesis. The experimental results show that the realism of the images produced by the synthetic network is very high, which is conducive to the development of alternative paths for image synthesis in computer graphics.

REFERENCES:

- I. I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio. Generative adversarial nets. In NIPS, 2014. 2, 6
- II. E. L. Denton, S. Chintala, A. Szlam, and R. Fergus. Deep generative image models using a Laplacian pyramid of adversarial networks. In NIPS, 2015. 2, 3, 6
- III. J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. IJCV, 2013. 2

- IV. J. Hosang, R. Benenson, P. Dollár, and B. Schiele. What makes for effective detection proposals PAMI, 2015
- V. A. Krizhevsky, I. Sutskever, and G. Hinton. ImageNet classification with deep convolutional neural networks. In NIPS, 2012
- VI. K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In ECCV. 2014. 2
- VII. R. Girshick. Fast R-CNN. In ICCV, 2015. 2,3,4,5,6
- VIII. S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In NIPS, 2015. 2,3
- IX. J. Dai, K. He, Y. Li, S. Ren, and J. Sun. Instance-sensitive fully convolutional networks. In ECCV, 2016. 2,3
- X. Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei. Fully convolutional instance-aware semantic segmentation. In CVPR, 2017.2,3,4
- XI. A. Shrivastava, A. Gupta, and R. Girshick. Training region based object detectors with online hard example mining. In CVPR, 2016. ,2
- XII. P. O. Pinheiro, R. Collobert, and P. Dollár. Learning to segment object candidates. In NIPS, 2015. 2,3
- XIII. A. Guzmán-Rivera, D. Batra, and P. Kohli. Multiple choice learning: Learning to produce multiple structured outputs. In NIPS, 2012. 5
- XIV. S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. In ICML, 2016. 1, 2, 3, 6
- XV. S. E. Reed, Z. Akata, S. Mohan, S. Tenka, B. Schiele, and H. Lee. Learning what and where to draw. In NIPS, 2016. 1, 2
- XVI. H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. Metaxas. StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks. In ICCV, 2017. 1, 2, 3, 6, 7
- XVII. I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio. Generative adversarial nets. In NIPS, 2014. 2, 6
- XVIII. Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of Wasserstein GANs. Advances in Neural Information Processing Systems (NIPS), 2017
- XIX. S. Ren, K. He, R. Girshick, and J. Sun. Fasterr-cnn: Towards real-time object detection with region proposal networks. In Advances in Neural Information Processing Systems, pages 91–99, 2015.
- XX. Ting Chen, Mario Lucic, Neil Houlsby, and Sylvain Gelly. On Self Modulation for Generative Adversarial Networks. In International Conference on Learning Representations (ICLR), 2019.
- XXI. V. Kumar Verma, G. Arora, A. Mishra, and P. Rai. Generalized zero-shot learning via synthesized examples. In CVPR, 2018
- XXII. Leonid Karlinsky, Joseph Shtok Sivan Harary and Eli Schwartz. RepMet: Representative-based metric learning for classification and few-shot object detection. In CVPR, 2019.
- XXIII. E. Schwartz, L. Karlinsky, J. Shtok, S. Harary, M. Marder, A. Kumar, R. Feris, R. Giryes, and A. M. Bronstein. - Encoder: an Effective Sample Synthesis Method for FewShot Object Recognition. NIPS, 2018. 3
- XXIV. T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature Pyramid Networks for Object Detection. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017. 4, 5
- XXV. Zhaojin Huang, Lichao Huang, Yongchao Gong, Chang Huang and Xinggang Wang. Mask Scoring R-CNN. In CVPR, 2019
- XXVI. Zhaojin Huang, Lichao Huang, Yongchao Gong, Chang Huang, Xinggang Wang. Mask Scoring R-CNN. In CVPR, 2018
- XXVII. J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. Deformable convolutional networks. In IEEE International Conference on Computer Vision, pages 764–773, 2017. 4, 6
- XXVIII. M. Pharr, W. Jakob, and G. Humphreys. Physically Based Rendering: From Theory to Implementation. Morgan Kaufmann, 3rd edition, 2016. 2
- XXIX. J. Portilla and E. P. Simoncelli. A parametric texture model based on joint statistics of complex wavelet coefficients. IJCV, 40(1), 2000. 4
- XXX. K. Sayood. Introduction to Data Compression. Morgan Kaufmann, 2012. 4