Group Report

Hongzhi Liu

July 18th, 2019

CONTENT

Review of Previous Work

1. Motivation

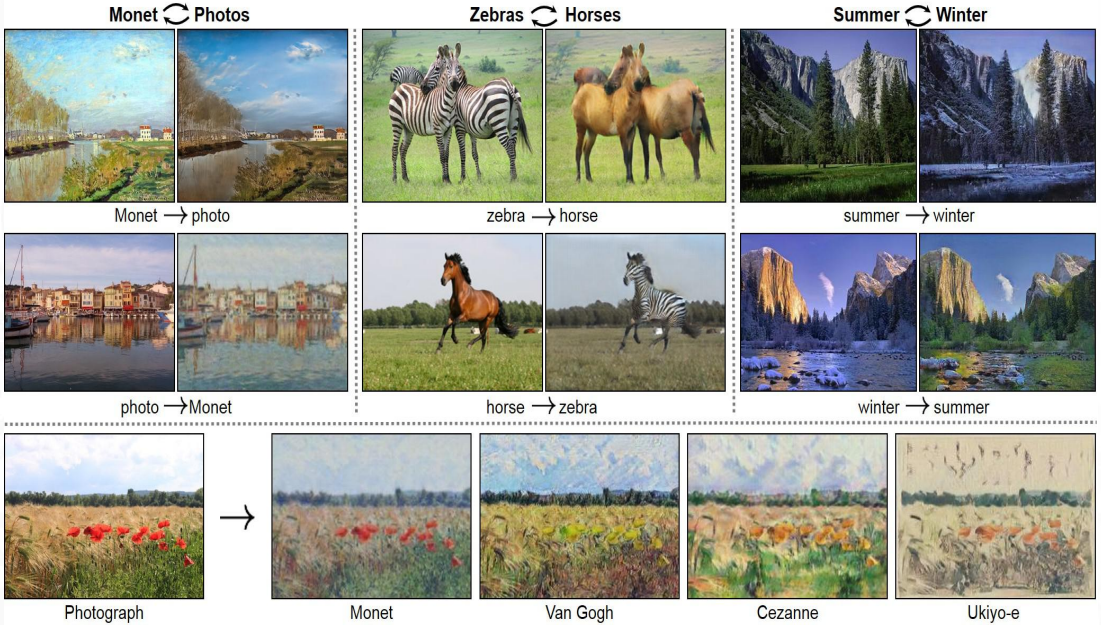

Figure 1. CycleGAN is a technique for image-to-image translation. **left:** shows the results of learning a transformation between two image distributions such as Monet and photo. **Right:** shows a gif about transformation between zebras and horses.

J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE international conference on computer vision, 2017.

1. Motivation

(a) Aerial photograph: x. (b) Generated map: Fx. (c) Aerial reconstruction: GFx

Figure 2. Details in x are reconstructed in GFx, despite not appearing in the intermediate map Fx.

Casey Chu, Andrey Z., Sandler M. CycleGAN, a master of steganography. arXiv preprint arXiv:1712.02950, 2017.

1. Motivation

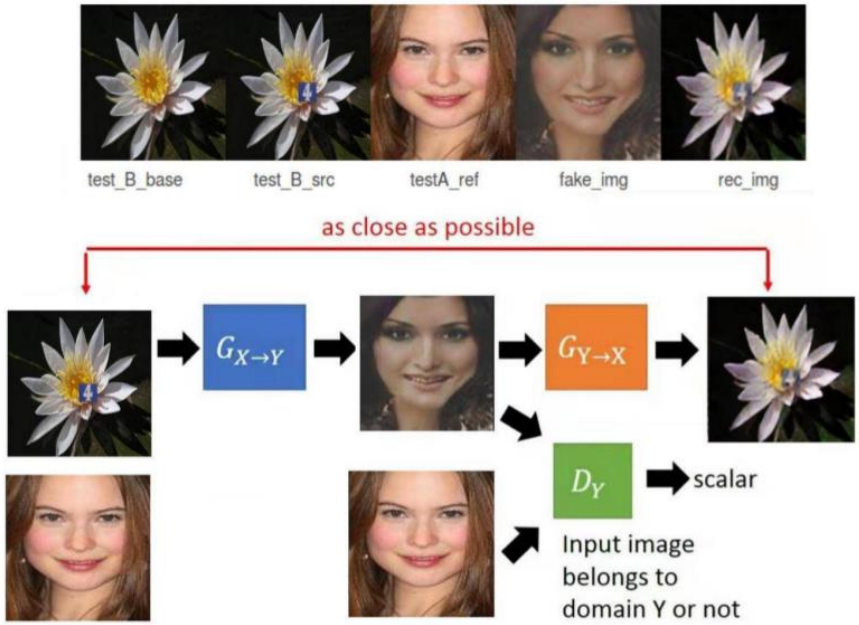

Figure 3. We want to design a network based CycleGAN network. The test_B_src contains a number as an encrypted message. The fake_img is composed of test_B_src and testA_ref image. We can get rec_img from the only one correct test_B_base image.

2. Method

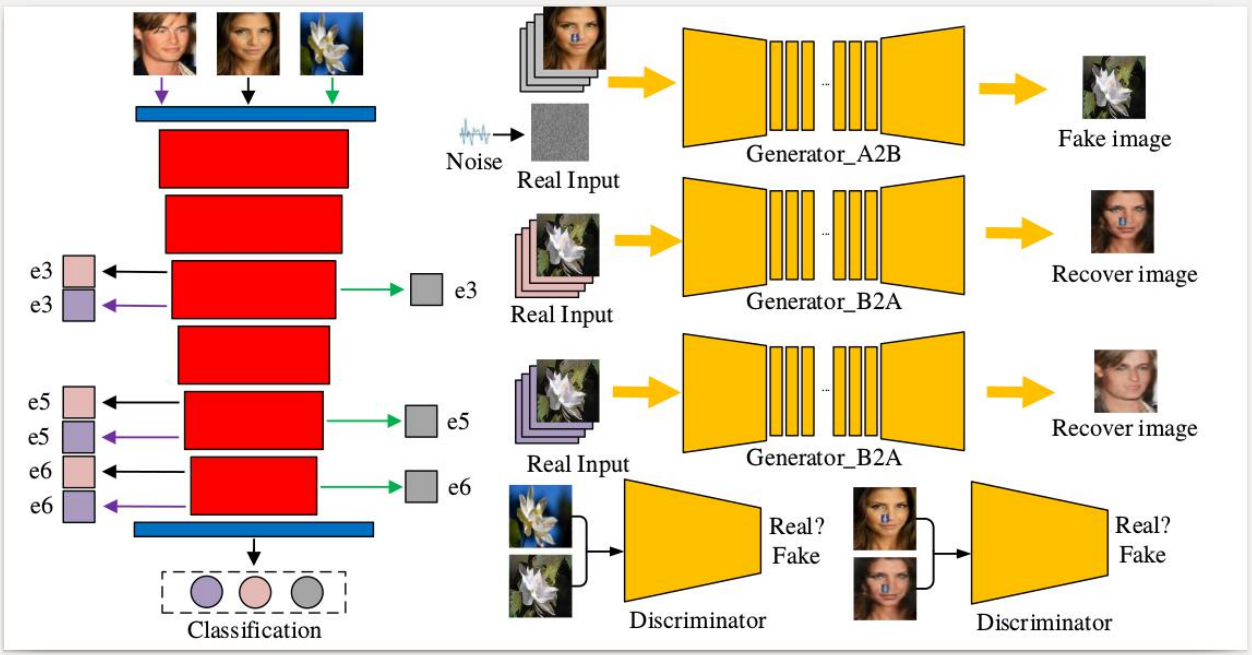

Figure 4. The preliminary network of our method. The classifier is able to classify human face and flower. Besides, the feature from different layers can be extracted to concate with Composite images which have number on human face.

Review of Previous Work

2. Method

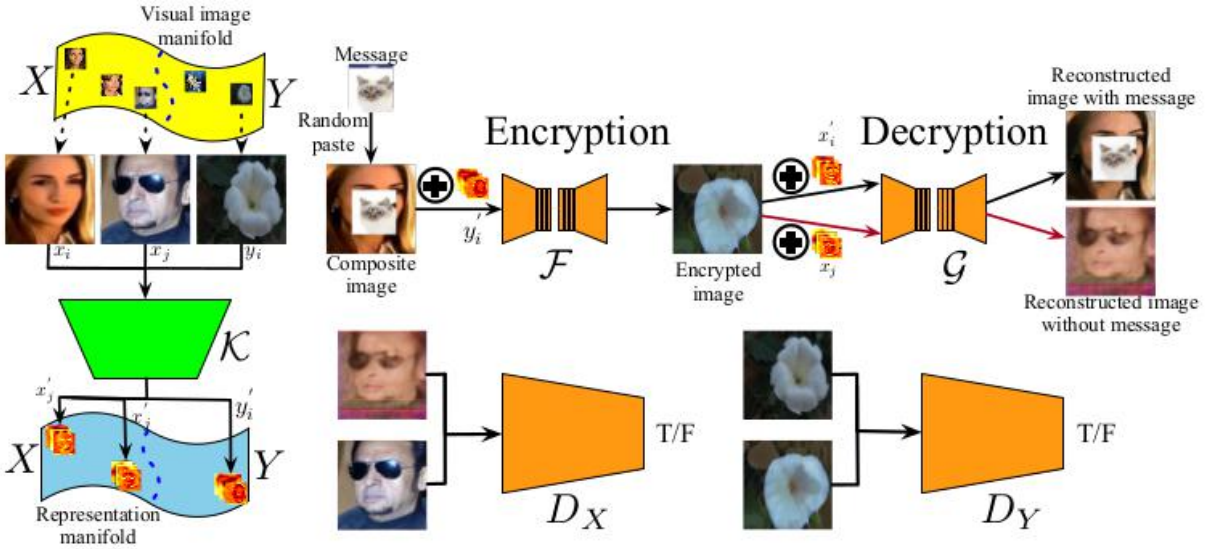

Figure 5. The network arcitecture of our **EncryptGAN**. The composite image includes human and number. The Encrypted images generated through the Encryption network. And reconstructed images generated from the Decryption.

Z. Zheng, H. Liu, Z. Yu, H. Zheng, et al. EncryptGAN: Image Steganography with Domain Transform. arXiv preprint arXiv:1905.11582, 2019.

2. Method

$$\mathcal{L}_{key}(\mathcal{F},\mathcal{G},x)=\sum_{i=1}^{N}\mathbb{E}_{x\sim P_{data(x)}}[||L_i(\mathcal{G}(\mathcal{F}(x)))-L_i(x)||_1+||L_i(\mathcal{F}(\mathcal{G}(x)))-L_i(x)||_1] \quad (1)$$

where $L_i(x)$ is the activation response of the i_{th} layer of the $\mathcal{K}$ with the input x; $\mathcal{F}(x)$ is the recovered image generated from $\mathcal{G}$; N is the number of layers we need for key generation. We choose $N=3$ layers (L_3,L_5 and L_6) in our experiments. The key matching loss aims to keep the representation similarity between x and $\mathcal{G}(\mathcal{F}(x))$ as well as y and $\mathcal{F}(\mathcal{G}(y))$.

Z. Zheng, H. Liu, Z. Yu, H. Zheng, et al. EncryptGAN: Image Steganography with Domain Transform. arXiv preprint arXiv:1905.11582, 2019.

2. Method

Information Loss: To ensure the information quality of the recovered image, we introduce the information loss for the part to 'cover' the plain information. The information loss $\mathcal{L}_{loc}$ is described as:

$$\mathcal{L}_{info}(\mathcal{F},\mathcal{G}) = \mathbb{E}_{x_{info}\sim P_{data(x_{info})}} \parallel \mathcal{G}(\mathcal{F}(x)) - x_{info} \parallel_1 + \mathbb{E}_{y_{info}\sim P_{data(\bar{y})}} \parallel \mathcal{F}(\mathcal{G}(y)) - y_{info} \parallel_1 \quad (2)$$

where x_{info} and y_{info} denote the local area including message information on both composite images and recovered images. The information loss aims to preserve the message information in the decrypted images.

Z. Zheng, H. Liu, Z. Yu, H. Zheng, et al. EncryptGAN: Image Steganography with Domain Transform. arXiv preprint arXiv:1905.11582, 2019.

3. Experiments

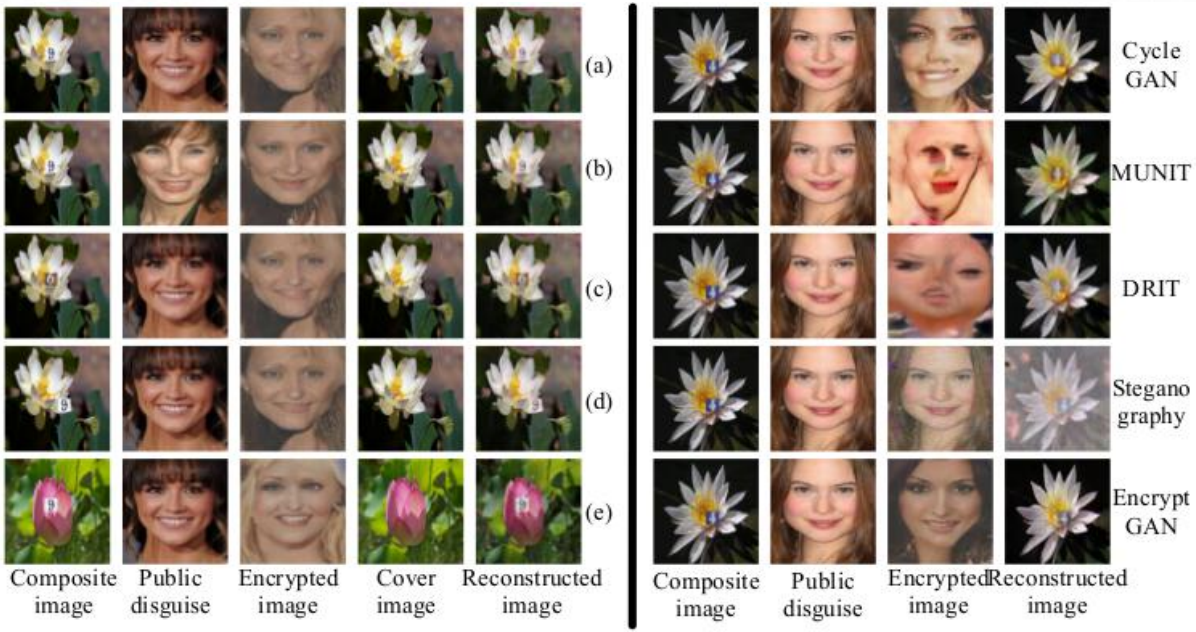

Figure 6. The visual comparison using different methods. **Left**. The effect under different circumstances. (a) and (b) are different with public disguise. We choose different digits in (a) and (c). Besides, position changes in both (a) and (d) in order to illustrate the randomness of encrypted information location. In (e), we achieved satisfying encryption on different cover images. **Right**. We compare several image translation algorithms with ours.

Z. Zheng, H. Liu, Z. Yu, H. Zheng, et al. EncryptGAN: Image Steganography with Domain Transform. arXiv preprint arXiv:1905.11582, 2019.

3. Experiments

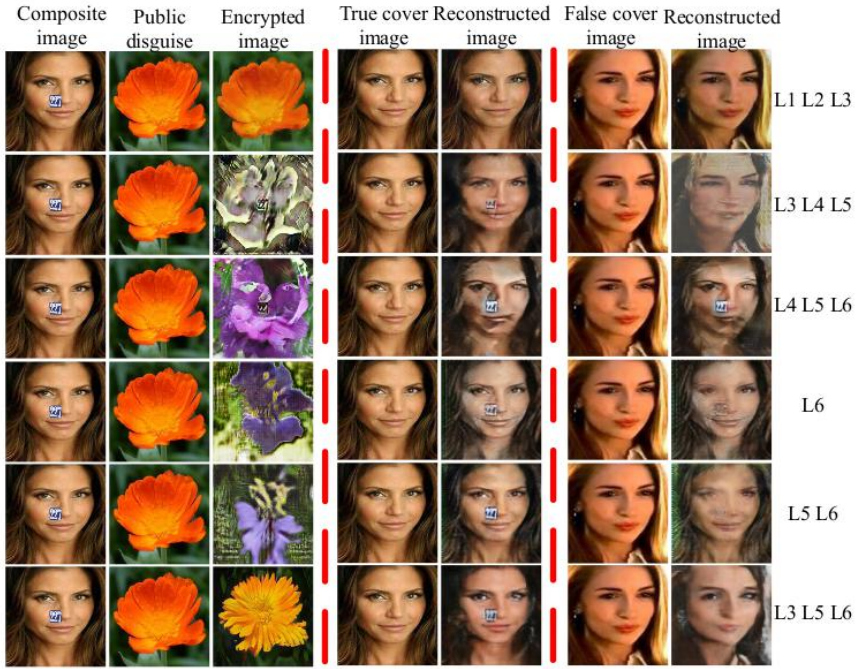

Figure 7. The ablation study of using different layer combinations for our **EncryptGAN**.

Z. Zheng, H. Liu, Z. Yu, H. Zheng, et al. EncryptGAN: Image Steganography with Domain Transform. arXiv preprint arXiv:1905.11582, 2019.

3. Experiments

Table 1. The quantitative comparison using differen layer combintions for image encryption task on flower↔face translation.

Methods	FID ↓	Encryption(LPIPS ↑ / MSE ↑)	MSE ↓	RMSE ↓	PSNR ↑	SSIM ↑	LPIPS ↓	Security(LPIPS ↓ / MSE ↓)
L_1, L_2, L_3	41.1943	0.4105 / 0.1257	0.1290	0.3541	7.8553	0.8197	0.4058	0.4249 / 0.1365
L_3, L_4, L_5	210.3436	0.3259 / 0.1169	0.0159	0.1131	18.5142	0.9694	0.1438	0.7116 / 0.1132
L_4, L_5, L_6	139.7629	0.3981 / 0.0857	0.0187	0.1299	16.8173	0.9715	0.2291	0.4182 / 0.0731
L_6	321.4196	0.3034 / 0.0647	0.0184	0.1337	16.3136	0.9782	0.2527	0.3634 / 0.0386
L_5, L_6	204.7240	0.3825 / 0.0613	0.0125	0.1094	18.1165	0.9860	0.2103	0.3934 / 0.0894
L_3, L_5, L_6(EncryptGAN)	107.9238	0.4141 / 0.0765	0.0139	0.1156	17.5992	0.9825	0.2414	0.5191 / 0.1379

Z. Zheng, H. Liu, Z. Yu, H. Zheng, et al. EncryptGAN: Image Steganography with Domain Transform. arXiv preprint arXiv:1905.11582, 2019.

3. Experiments

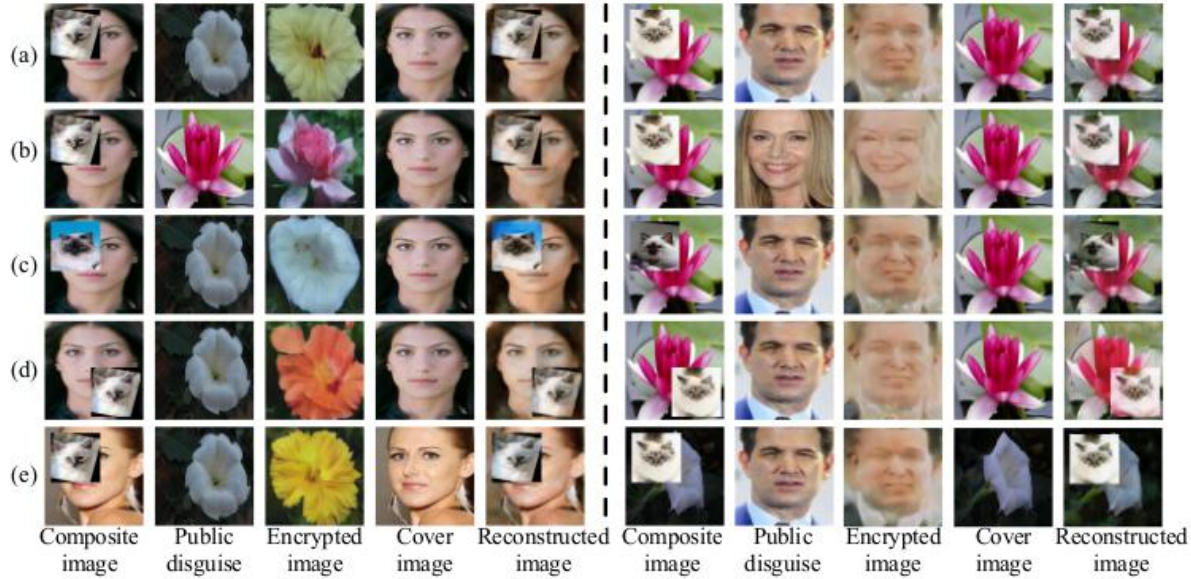

Figure 8. Cover images with complex information such as cat and dog images.

Z. Zheng, H. Liu, Z. Yu, H. Zheng, et al. EncryptGAN: Image Steganography with Domain Transform. arXiv preprint arXiv:1905.11582, 2019.

4. Conclusion

We proposed a domain translation framework that achieves mage asymmetric encryption by combining the public-key crypography system which is the first such architecture using GAN.

Our EncyptGAN can achieve a semantic level message steganography and perform better than current image-to-image translation and image steganographic methods. We devised the ablation studies for our method and performed a comprehensive and specific analysis.

Z. Zheng, H. Liu, Z. Yu, H. Zheng, et al. EncryptGAN: Image Steganography with Domain Transform. arXiv preprint arXiv:1905.11582, 2019.

Preparation for URPC 2019

1. Competition Rules

本竞赛参考了ILSVRC2015 Object Detection任务的评测方法，采用平均准确率（mAP）作为评价指标。（详细说明请参考http://image-net.org/challenges/LSVRC/2015/index#maincomp）给定目标框真值BG与目标种类真值CG，假设算法预测的目标框为B，预测的目标种类为C。当C与CG一致并且目标框BG与B的重叠率(IOU)大于某一阈值时判定该预测为一次准确检测。目标框的重叠率计算方式为:

IOU(B,BG) = (B intersection BG) / (B union BG)

对于大小为m*n的真值目标框，阈值计算方法为

thr = min(0.5, m*n / [(m+10)*(n+10)])

不满足上述条件的检测结果（种类不一致或目标框重叠率低于阈值）将被判为误检。对同一目标的重复检测也将被判为误检。

参赛算法的预测结果将以一个txt格式的文件提交。文件的每一行对应一个检测到的目标，格式如下：

<image_id> <class_id> <confidence> <xmin> <ymin> <xmax> <ymax>

其中image_id为测试图片的id号（列于devkit/data/test文件中）, class_id 为物体的种类（参见devkit/data/meta_data.mat）, confidence为算法对于这一预测的置信度，xmin ymin为目标框左上角点坐标，xmax ymax为目标框右下角点坐标。

用于评测算法的MATLAB程序位于devkit/evaluation/eval_detection.m。为了便于理解，我们提供了示例程序devkit/evaluation/demo_eval_det.m。该程序对val集上的预测结果文件devkit/evaluation/demo_val_pred.txt进行评测。

2. Object Detection Algorithm

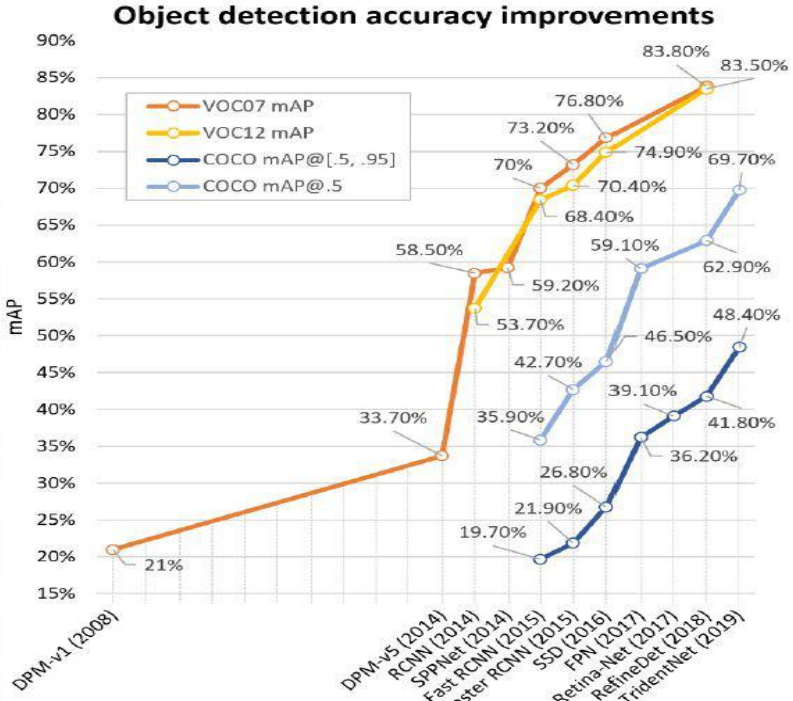

Figure 9. The accuracy improvements of object detection on VOC07, VOC12 and MS-COCO datasets. Detectors in this figure: DPM-v1,DPM-v5, RCNN, SPPNet, Fast RCNN, Faster RCNN, SSD, FPN, Retina-Net, RefineDet and TridentNet.

2. Faster RCNN & RPN

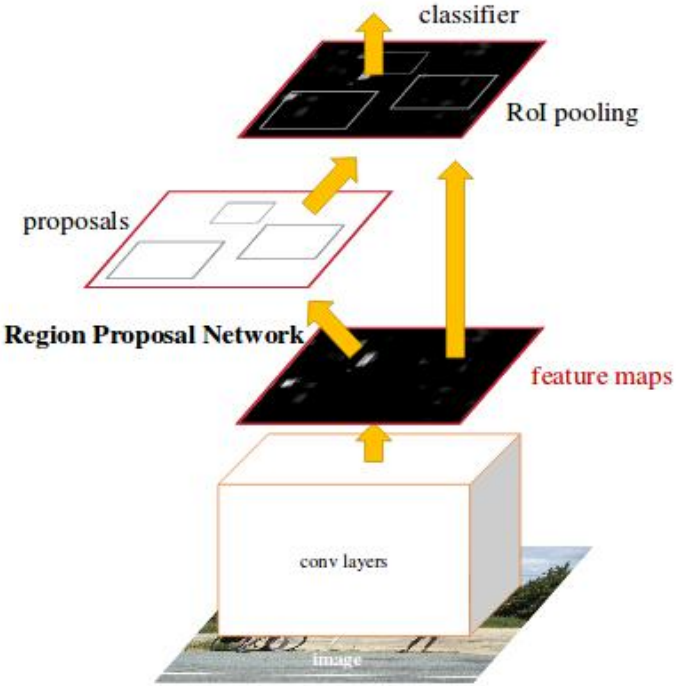

Figure 10. Faster R-CNN is a single, unified networkfor object detection. The RPN module serves as the 'attention' of this unified network.

2. Faster RCNN & RPN

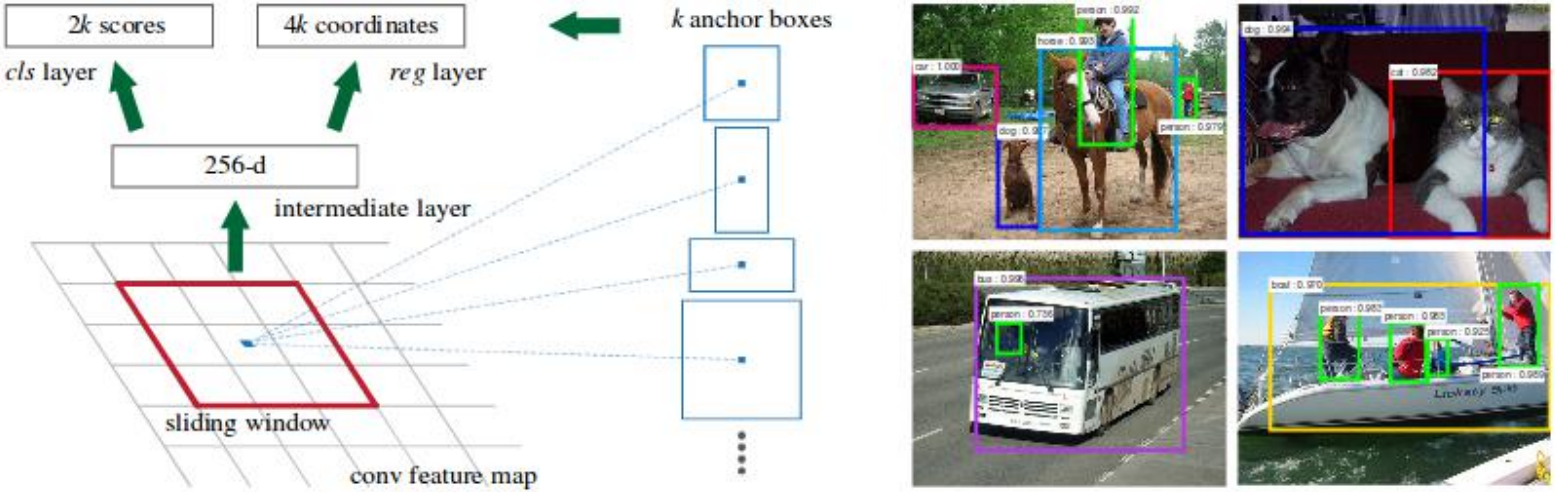

Figure 11. **Left**: Region Proposal Network (RPN). **Right**: Example detections using RPN proposals on PASCAL VOC 2007 test. The method detects objects in a wide range of scales and aspect ratios.

3. Experiment

Table 2. The Faster RCNN is trained with different classification network such as VGG16, res101 and res152. We can get the best mAP using restoration datasets.

Classifer	Threshold	mAP
VGG16	0.3	0.8356
res101	0.5	0.8422
res152	0.7	0.8464
	0.5	0.8780 (Restoration)

3. Experiment

```
henry@henry-System-Product-Name:
im_detect: 317/324 0.231s 0.003s
im_detect: 318/324 0.231s 0.003s
im_detect: 319/324 0.231s 0.003s
im_detect: 320/324 0.231s 0.003s
im_detect: 321/324 0.231s 0.003s
im_detect: 322/324 0.231s 0.003s
im_detect: 323/324 0.231s 0.003s
im_detect: 324/324 0.231s 0.003s
Evaluating detections
Writing holothurian VOC results file
Writing echinus VOC results file
Writing scallop VOC results file
Writing starfish VOC results file
VOC07 metric? Yes
AP for holothurian = 0.9869
AP for echinus = 0.9998
AP for scallop = 0.5348
AP for starfish = 0.9922
Mean AP = 0.8784
~~~~~~~~
Results:
0.987
1.000
0.535
0.992
0.878
~~~~~~~~
```

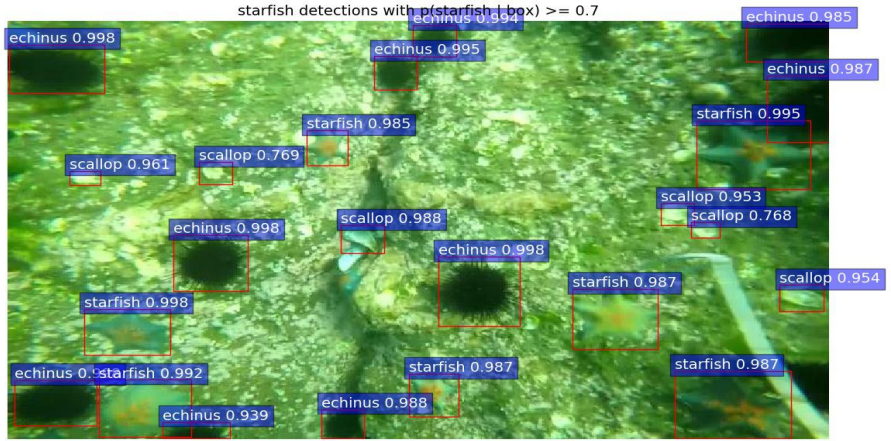

Figure 12. We visualize the test results with bounding box and confidence value. Some rocks may easily mistaken for scallops.

Further Research

1. Additional Experiments for different networks and algorithm

2. Looking for better advice and methods

Q & A