國立交通大學

電控工程研究所

碩士論文

對抗式生成網路於機器人實際場域應用

Generative Adversarial Networks for Real-robot Missions

研 究 生：陳品維

指導教授：王學誠 博士

中 華 民 國 一 百 零 八 年 十 二 月

對抗式生成網路於機器人實際場域應用

Generative Adversarial Networks for Real-robot Missions

研 究 生：陳品維
指導教授：王學誠

Student: Pin-Wei Chen
Advisor: Hsueh-Cheng Wang

國 立 交 通 大 學
電 控 工 程 研 究 所
碩 士 論 文

A Thesis
Submitted to Institute of Electrical Control Engineering
College of Electrical and Computer Engineering
National Chiao Tung University
in partial fulfilment of the requirements
for the Degree of
Master
in

Electrical Engineering

August 2019

Hsinchu, Taiwan

中 華 民 國 一 百 零 八 年 十 二 月

對抗式生成網路於機器人實際場域應用

學生：陳品維　　指導教授：王學誠 博士

國立交通大學 電控工程研究所

摘　　要

在深度學習的高度發展以及人工智慧的浪潮下，電腦視覺的技術以及應用層面皆大幅度地的成長。然而隨著生成模型 (Generative Model) 的出現，如今電腦不僅僅能做出基本的影像處理和影像辨識等任務，它還能夠跟人類一樣，運用自己的智慧去" 創造" 影像。其中對抗式生成網路 (Generative Adversarial Network) 更是在人們大量的研究及開發下，產生很多的網路模型變形以及不同的應用，像是圖畫風格轉換、人臉表情轉以及文字產生圖像等。但卻鮮少有人將對抗式生成網路運用於機器人實際場域中，讓它們試著去取代其他方法。因此本論文將會根據兩個在運用現今常用方法仍會遇到的挑戰以及困境的機器人任務中，試著基於兩個著名的對抗式生成網路: Pix2Pix [1] 以及 CycleGAN [2]，提出兩個新的對抗式生成網路: FCN-Pix2Pix 以及 SSIM-CycleGAN，並將這兩個改良過後的對抗式生成網路，應用在這兩個在機器人任務裡，並解決所遇到的難題以及困境: **1) 機器人視覺之影像像素級辨識 (Semantic Segmentation) 2) 於實際場域運用虛擬環境資料集 (Virtual Dataset from Sim to Real)**。本論文也會針對這兩個議題，把基於 GAN 所提出的方法與其他新穎的方法做比較，並於實驗結果中顯示出本論文所提出的方法之優勢。

Generative Adversarial Networks for Real-robot Missions

Student: Pin-Wei Chen Advisor: Dr. Hsueh-Cheng Wang

Institute of Electrical Control Engineering
National Chiao Tung University

Abstract

Leveraging highly developed deep learning and artificial intelligence, computer vision technology and applications reached new levels. Computers can now not only perform image processing, classification, and object detection, but also can ”create” images similarly to humans, due to generative model developments. In particular, the generative adversarial network (GAN) provides many architectures and applications, such as image style transfer, human face generation, image generation from text, etc. However, there has been little study regarding applying GAN to real-robot missions to replace and improve other approaches. Therefore, this work proposed two GANs: FCN-Pix2Pix and SSIM-CycleGAN, based on Pix2Pix and CycleGAN respectively, and implemented them for two real-robot missions which still face some challenges with modern solutions: semantic segmentation and virtual dataset from sim to real. The proposed approaches were also compared with current state-of-the-art approaches, verifying significant advantages for the proposed methods.

Acknowledgement

首先我要感謝我的指導教授王學誠老師，他在我就讀研究所的階段，給予了我非常多的機會，不論是比賽、會議、展覽等等，都讓我從中獲得到非常豐富的經驗以及歷練，這也讓我有一個非常難忘且精彩的研究所時光。同時，王學誠老師也提供了整個實驗室非常豐富的資源及設備，讓我們能夠在這樣的環境中，擁有極佳的研究環境去精進自我。

另外我也要感謝口試委員楊谷洋教授、帥宏翰教授，正因有了他們對於我論文的指教與建議，我才得以提升本論文的完整性及專業度。我也很感謝這一路走來所有指導過我的老師，他們在我求學的路上，教會了我學業以及做人處事的種種道理，讓我能夠一直對著生活有著正面的態度及積極度。

在實驗室的這段期間，所有與我一同相處共事過的實驗室同學們及助理們，也是這一路走來非常照顧我及支持我的人。我們在比賽、研究及課業中，總是一同努力一同鼓勵彼此，就算是低潮時期，也會互相勉勵，讓彼此都對生活一直保持著熱情，並一路堅持奮鬥到現在。也因爲在這樣充滿活力以及溫暖的實驗室氛圍下，我才如此的珍惜我的碩士時光，因此我想在此感謝實驗室的每一位成員。

最後，我最要感謝我的家人們。謝謝爸爸媽媽從小到大的養育及教誨，教導我正確的人生觀，並且讓我一路上有著很好的教育，就算到了大學、研究所階段來到新竹念書，他們還一直在高雄支持著我，成爲我最好的後盾，在此我要深深地感謝我的爸爸媽媽和所有的家人們。

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivations and Challenges

Computer vision has advance greatly, leveraging machine learning, deep learning, and artificial intelligence developments to be applied in many fields. Consequently, computer vision has becomes much faster with better prediction accuracy compared with traditional slow and relatively poor accuracy image processing, achieving comparability with human performance.

The development of generative models has allowed computers to not only realize complex image processing, but also to create images directly, behaving similarly to humans. The generative adversarial network (GAN) is currently the most popular network, being widely used for many applications, such as image style transfer, generating human facial images, face aging, etc. The full list of successful applications is impressive, but most are computer vision aspects, rather than real-robot applications. Therefore, I propose to apply powerful generative adversarial network architecture for robotics in real-robot missions.

Studying two recent real-robot missions, DARPA Subterranean (SubT) Challenge [4] and the robot arm pick-and-place [5] by NCTU ARG Lab [6], I found some disadvantages that could be improved by GAN. Therefore, I propose to use GAN to address two essential issues for real-robot missions: semantic segmentation and virtual dataset from simulation to real.

- Semantic Segmentation

The DARPA SubT Challenge [4] seeks novel approaches for autonomous rapidly mapping, navigation, and artifact searching in unknown underground environments during time-sensitive combat operations. The artifact search requires the robot to automatically detect artifacts and obtain their precise position. Consequently, I need to build a robust and powerful image semantic segmentation solution for this task.

Semantic segmentation is an important but difficult robot vision technology to give every pixel a predicted class, rather than only providing two-dimensional (2D) bounding box predictions for an image. However, common semantic segmentation networks generate predictions with considerable noise and checkerboard artifacts. I propose to use GAN to solve these problems.

- Virtual Dataset from Simulation to Real

Many studies have tried to fully automate factory logistics and warehousing, enhancing process speed and reducing labor costs. The NCTU ARG Lab [6] considered warehouse robotic pick-and-place systems with the aim to automatically pick-and-place [5] all products. Their proposed system detected object location and assigned a category, and then let the robot arm grasp and move it to the appropriate shelf following appropriate pose.

Object detection is mostly achieved by data-driven deep learning approaches, which requires considerable labeled data for training. The required dataset sizes can raise concerning money and time costs during data collection. This problem has begun to be addressed by generating data in appropriate virtual environments, providing tens of thousands of data in very short time, covering many different scenes. However, deep neural network models trained with virtual data generally provide poor performance for real environments. Therefore, I propose to build a GAN system to convert the virtual dataset from simulation to real environment, such that an object detection deep neural network trained with the new dataset can be used in real environments.

1.2 Contributions

This work aims to develop GAN for robotics applications. I propose GAN based methods to solve semantic segmentation and virtual dataset from simulation to real problems. The proposed approaches will be compared with current state-of- the-art methods to verify robustness and capability.

- FCN-Pix2Pix

The proposed FCN-Pix2Pix semantic segmentation system was based on Pix2Pix GAN [1] to improve accuracy while reducing noise and checkerboard artifacts for image pixel-wise classification.

- SSIM-CycleGAN

The proposed SSIM-CycleGAN system generated a Sim2Real dataset from a virtual dataset using CycleGAN [2] to create a suitable dataset such that an object detection network trained on it could be used in real environments.

1.3 Thesis Architecture

Section 1 discusses the underlying motivations, challenges, and contributions of this paper, and then Section 2 reviews relevant previous reported studies, introducing GAN [7], Pix2Pix [1], and CycleGAN [2] networks that are used subsequently. Sections 3 and 4 discuss the experiments for semantic segmentation and virtual dataset from simulation to real, respectively. Finally, Section 5 summarizes and concludes the paper, and discusses potential future research directions.

Chapter 2

Literature Review

2.1 Generative Adversarial Network

Generative adversarial networks (GANs) [7] generally comprise a generator and discriminator, both of which are neural networks. The core GAN concept is to estimate the generator, G, by the discriminator, D, using an adversarial process. We simultaneously train G and D, using G to capture the data distribution, and D to estimating the probability a given sample comes from the training data X or the generator G. We can consider the GAN architecture as a minimax two-player game, where G is trying fool D that $G(z)$ is same as X, and D is trying show $G(z)$ and X are different.

Training the GAN model can be divided into two steps. First, we maximize the probability of D making a mistake for G, i.e., we train G to minimize $log(1-D(G(z)))$, and D to maximize $log(1-D(G(z)))$ and $log(D(x))$. Therefore, the goal of the GAN is to optimize 2.1.

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[logD(x)] + \mathbb{E}_{z\sim p_z(z)}[log(1-D(G(z)))] \tag{2.1}$$

GAN is one of the most powerful constructs from machine learning in the last decade, with many extended applications and modifications being actively developed, including different generators and discriminators as well as the underlying adversarial architecture. For example, [8] demonstrated StackGAN, using a series of GANs to generate realistic images from textual descriptions; [9] used GANs to generate new images of human models in different poses, and [10] employed GANs to generate facial images at different ages. The current study modified two GAN architectures, Pix2Pix [1] and CycleGAN [2] for semantic segmentation and virtual dataset from simulation to real, respectively. The underlying GAN systems are discussed in the following sections.

2.2 Pix2Pix

Pix2Pix [1] provides image-to-image translation with conditional adversarial networks, comprising a generator, G, and discriminator, D. The Pix2Pix goal provide the translation from source to target images, which are paired. For training, G takes the source image as conditional input and provides a generated image. D then learns how to tell whether a given image is generated image or the original target. Consequently G learns how to generated images that can fool D they are target images. The original Pix2Pix study [1] demonstrated translation between object edges to realistic images, and also translation between labeled masks to the actual street scene. Therefore, I modified Pix2Pix (FCN-Pix2Pix) to translate an image into a heatmap for semantic segmentation.

2.3 CycleGAN

CycleGAN [2] is an unpaired image-to-image translation using cycle consistent adversarial networks. CycleGAN comprises four models: G_{AB}, G_{BA}, D_A and D_B. G_{AB} and G_{BA} are generators, where G_{AB} takes an image from domain A as input, and tries to generate a fake domain B image; and G_{BA} generates a fake domain A image from domain B. D_A and D_B discriminate whether an image is from domain A or B, respectively. The CycleGAN goal is not only to transform an image from domain A to B but also from domain B to A, hence regularizing the generator. Since CycleGAN is an unpaired image-to-image network, we can train the network to transforming from domain A to B and domain B to A without requiring paired images between domains A and B. This is the key point for choosing CycleGAN as the underlying architecture for the proposed SSIM-CycleGAN system to convert a virtual dataset from simulation to real.

Chapter 3

Semantic Segmentation

3.1 Introduction

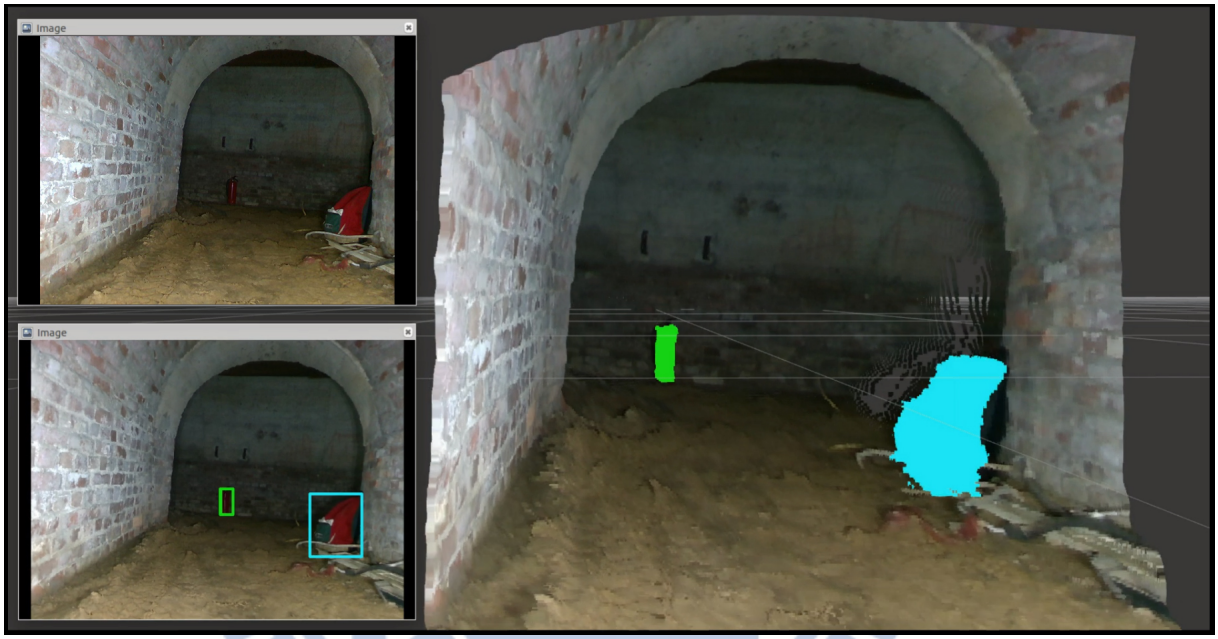


Figure 3.1: Subterranean (top left) image with artifacts, (lower left) artifact object detection by SSD [3], and (right) after semantic segmentation by FCN-Pix2Pix

I need to build a semantic segmentation model to find precise target artifact positions for the DARPA SubT Challenge artifact search. In this mission, I have to get a precise position of the artifact, and the semantic segmentation enables us to align the prediction mask to the depth image from the RGB-D camera. Therefore, we can use this information to compute the average depth of the artifact, and then get a more precise position compared with the bounding box result. This is because the bounding box result cannot tell us which pixel in the bounding box we should take into account when computing the depth, so I choose the semantic segmentation approach in the artifact search mission in DARPA SubT Challenge.

Predictions generated by most semantic segmentation deep neural networks include considerable noise and checkerboard artifacts [11] due to the unknown severe subterranean environ-

ments with varied lighting conditions, mist, dust, and fog, which cause many problems for robot vision to complete artifact searching.

This section discusses the proposed approach and shows achieves better performance than several state-of-the-art approaches as well as artifact searching from the University of Pennsylvania [12], who participated in the 2019 SubT Challenge.

3.2 Related Work

3.2.1 Object Detection

Modern vision based missions mostly use data-driven deep learning approaches, where deep neural network models are trained to provide predictions using large training datasets. Object detection has already been achieved many times using deep learning networks, such as the single shot multi-box detector (SSD) [3], YOLO [13], YOLO-V3 [14], R-CNN [15], fast-RCNN [16], and faster-RCNN [17]. These object detection networks all take an image as input and output 2D bounding boxes with predicted categories.

3.2.2 Semantic Segmentation

In contrast with object detection networks, which only provide 2D bounding boxes with categories, semantic segmentation networks give each input image pixel a predicted output class, making the DARPA SubT Challenge artifact search a better result for detecting the artifact poses. The fully convolutional network (FCN) [18] comprises a series of fully convolutional layers, including convolution and deconvolution layers. FCNs take arbitrary sized input images, passes them through the convolution layer, and then generates corresponding sized output with pixel-wise predicted labels for the input image. Most FCNs use VGG16 [19] as their encoder, which won the 2014 Large Scale Visual Recognition Challenge (ILSVRC2014) [20], hence provide excellent performance for pixel-wise image classification tasks. Therefore, I adopted an FCN based GAN approach to build the semantic segmentation network.

U-Net [21] has the same underlying architecture as FCN, but crops and concatenates convolution (down-sampling) layers to deconvolution (up-sampling) layers to extract more features

from different aspects. This approach makes U-Net architecture look U-shaped, which is the basis for "U-Net". U-Net is mostly applied in GANs architecture, however, it does not use VGG [19] for feature extraction, because U-Net is mainly for biomedical image segmentation. Therefore, it is not considered in the GAN approach for semantic segmentation task in this work.

3.3 Method

3.3.1 Network Architectures

I chose FCN as the network backbone due to its performance for pixel-wise image classification. In principle an FCN can immediately solve semantic segmentation problems, but in severe environments, FCN predictions contains significant checkerboard artifacts [22], [11] and noise. Consequently, I applied FCN to GAN architecture which is based on the Pix2Pix [1]. The GAN discriminator enables the network to consider the whole image when generating the mask prediction. Thus, network prediction can reduce checkerboard artifacts and noise, and consequently produce more precise predictions.

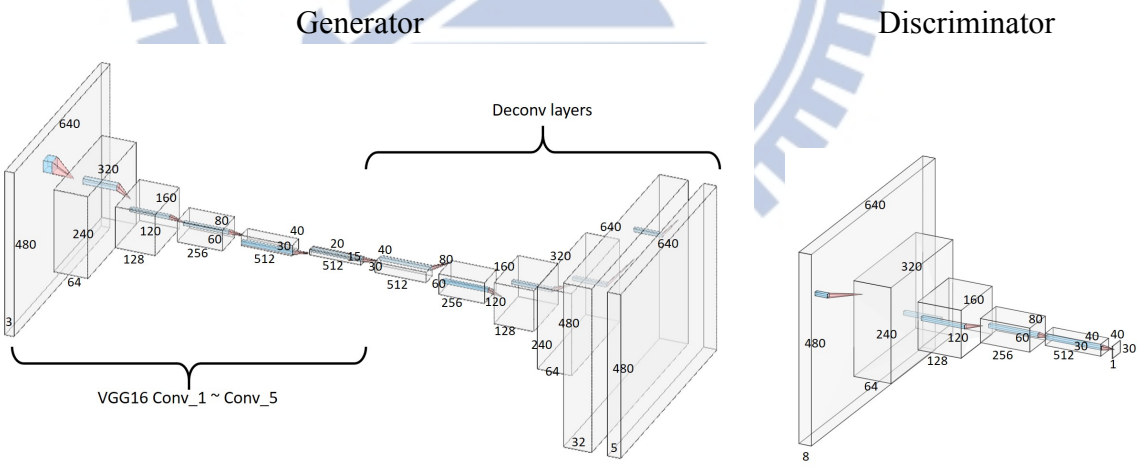


Figure 3.2: GAN components (left) generator, an FCN [18] model, and (right) discriminator, in this case a patch discriminator.

Previous Pix2Pix [1] studies uses U-Net as the generator, producing a one channel grayscale or three channel RGB image. However, since the current problem is a classification task, I used FCN as the generator Fig. 3.2, producing an n-channel feature map (heatmap) where n represents the number of classification categories.

The FCN generator architecture can be divided down and up-sampling components. An input RGB image (480×640×3) is first down-sampled through the VGG16 network’ top 5 convolution layers. The output tensors are then passed through the up-sampling network, comprising deconvolution (transposed convolution) layers, and the tensor is up-sampled step by step until the height and width are the same as the input image. Final FCN output is a five channel heatmap (480×640×5), i.e., every pixel has 5 values representing the probability of being the corresponding class: extinguisher, backpack, drill, survivor, and background. The predicted mask is set to be the class index with maximum probability, hence the final predicted mask has dimension 480×640×1.

Following Pix2Pix [1], I use the convolutional PatchGAN classifier as the discriminator, comprising six convolution and max-pooling layers. Input is a 480×640×8 tensor, the concatenation of the RGB image (480×640×3) and generated heatmap (480×640×5). Typical discriminators pass the input tensor through the convolution layers and classify at the last layer, providing a one dimensional 1 or 0 as the final result. Although, the PatchGAN employed in this work also passes the input tensor through the convolution layers, it produces 30×40 = 1200 patches in the last layer, where each patch is classified as 1 or 0. Thus, the discriminator produces many (true or false) classification results, rather than only a single result, greatly improving discriminator supervision capability.

3.3.2 Training

I built the FCN-Pix2Pix using the defined FCN generator and PatchGAN discriminator, with system architecture as shown in Fig. 3.3. The upper figure shows that RGB images pass through the generator and generate heatmaps. Then the heatmap is concatenated with the corresponding RGB image.

The discriminator learns how to distinguish generated images from ground truth labeled masks over many iterations, which simultaneously forces the generator to generate better prediction masks.

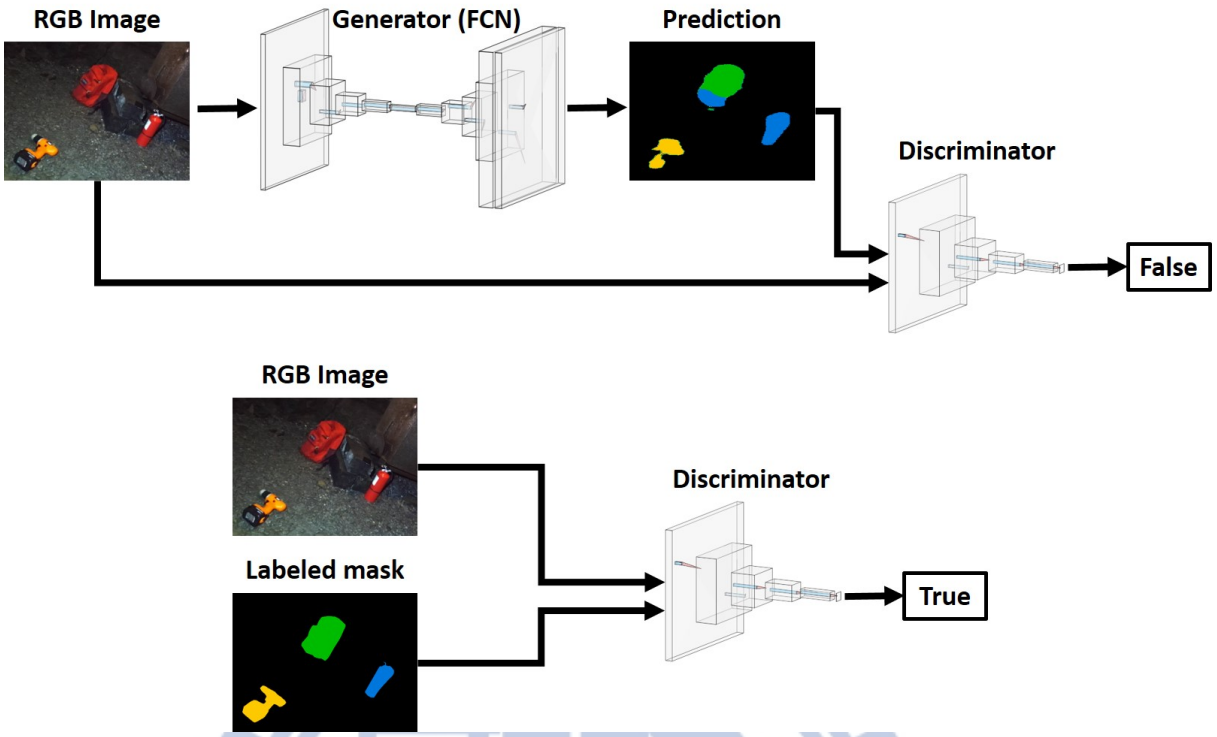


Figure 3.3: Proposed FCN-Pix2Pix architecture: (upper) generator and (lower) discriminator training procedures. Note that the upper and lower discriminators are the same. The concatenation of the RGB image (480×640×3) and generated heatmap is passed through the discriminator. This step enables the generator learning to confuse the discriminator (i.e., outputting True for a generated image. However, the discriminator simultaneously learns to classify generated and original images, the crux of the "adversarial" step. (lower) discriminator input, i.e., concatenated RGB image and corresponding ground truth labeled mask, which the discriminator learns to classify as True.

3.4 Experiments

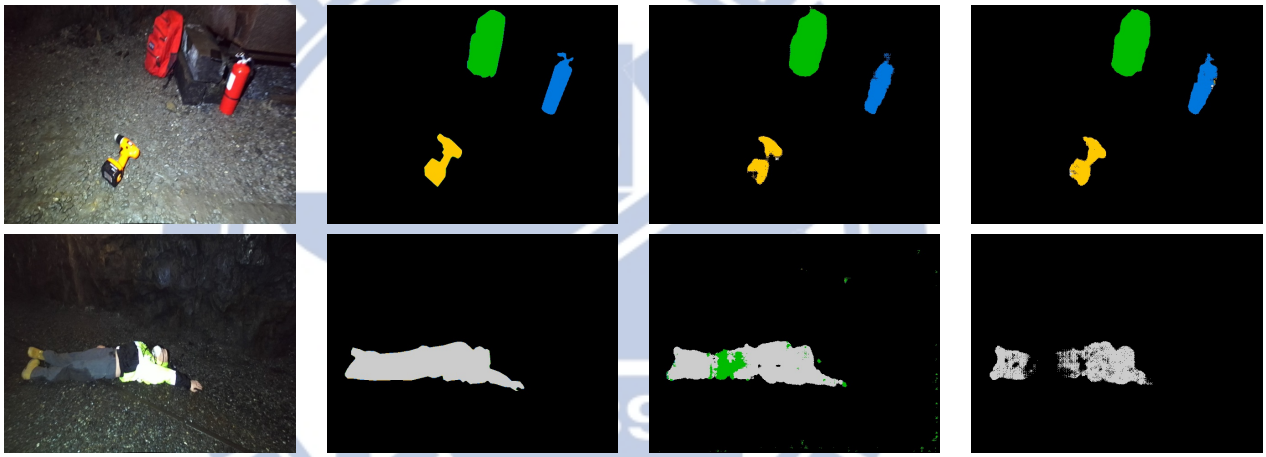
3.4.1 PST900 Dataset Introduction

The University of Pennsylvania [12] also participated in the DARPA SubT challenge and collected the SubT data into the PST900 dataset. Therefore, I used the PST900 dataset as the benchmark for this experiment. The dataset comprised 894 synchronized and calibrated RGB and thermal image pairs with pixel-level human annotations across four distinct classes: drill, extinguisher, backpack, survivor, and background.

3.4.2 Experiment Design

This experiment used RGB images from the PST900 training dataset to train several state-of-the-art semantic segmentation models, including ERFNet [23], MAVNet [24], UNet [21], Fast-SCNN [25], PST-segnet [12], FCN [18], and the proposed FCN-Pix2Pix. The trained models were then evaluated on the PST900 testing dataset using evaluation metrics image-level mask intersection over union (IoU), i.e., union overlap area for each class ($IoU = \frac{area\ of\ overlap}{area\ of\ union}$); and mean IoU (mIoU) over all classes.

3.4.3 Results and Discussions



(a) (b) (c) (d)

Figure 3.4: Normal network prediction outcomes for two images: (a) input RGB and (b) labelled images from PST900 dataset [12], (c) FCN predicted mask, (d) FCN-Pix2Pix predicted mask.

Figure 3.4 shows that FCN and FCN-Pix2Pix predict well for the PST900 testing dataset, but Figure 3.5 shows there are many differences between them. FCN predicted masks have considerable noise and checkerboard artifacts [11], which are typical disadvantageous for up-sampling (deconvolutional layers), whereas FCN-Pix2Pix prediction considerably reduce checkerboard artifacts and noise.

Implementing the semantic segmentation model in GAN architecture provides several benefits due to the discriminator, because typical semantic segmentation models only reduce L1 or L2 loss between predicted and labeled masks. Prediction results with blurring, noise, and checkerboard artifacts are difficult for FCN to detect since it can only calculate L1 or L2 loss

pixel by pixel, making it difficult to learn how to avoid these phenomena during backpropagation. In contrast, the discriminator helps detect masks with noise, checkerboard artifacts, and blurring as fake rather real images because the FCN-Pix2Pix network considers the whole image aspect. This was the core reason I implemented the semantic segmentation FCN model in GAN architecture.

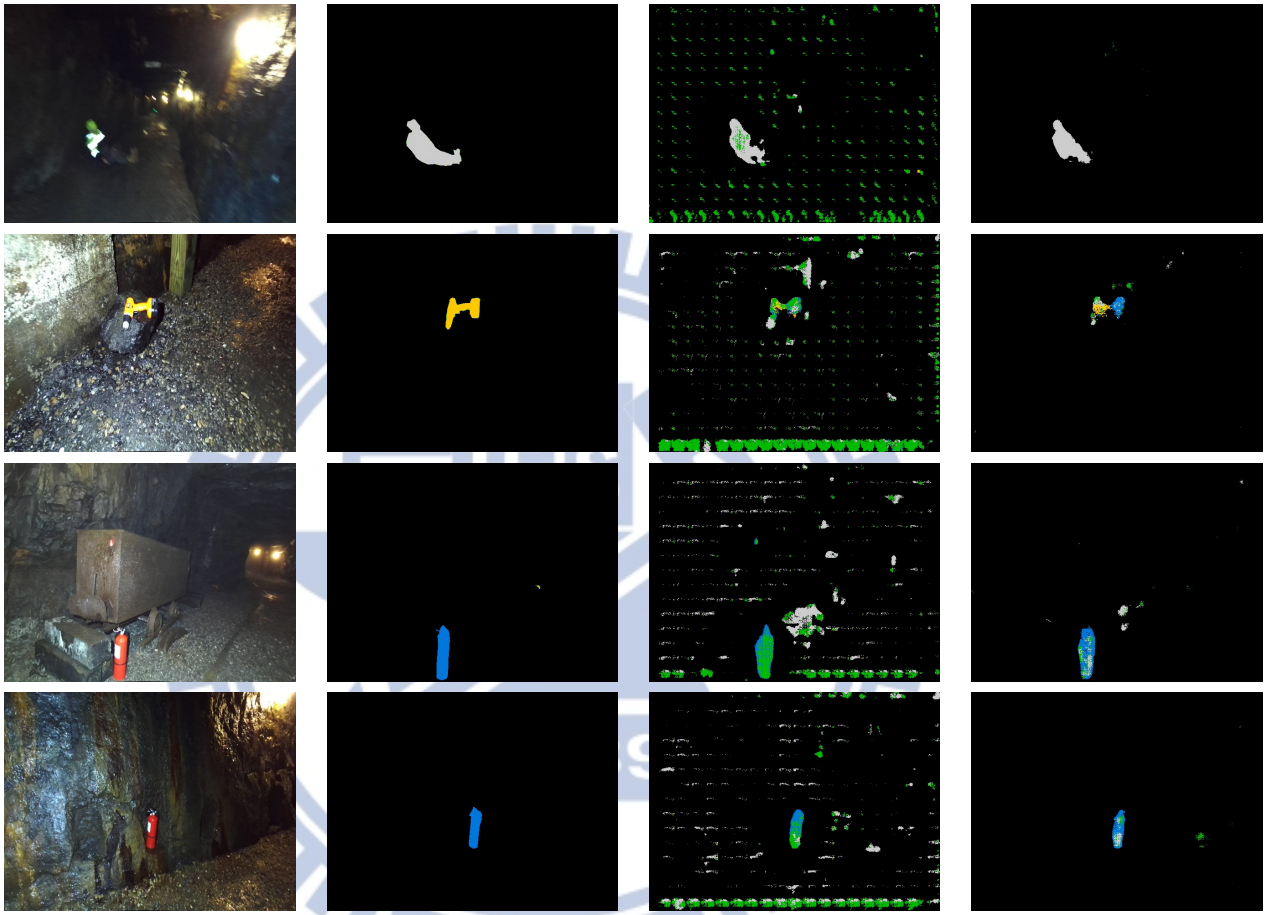


Figure 3.5: Example prediction outcomes: (a) RGB and (b) labeled images from the PST900 dataset [12]; (c) FCN predicted mask showing classic checkerboard artifacts; and (d) FCN-Pix2Pix predicted mask, completely free of checkerboard artifacts

Table 3.1 shows that FCN and FCN-Pix2Pix have superior performance compared with the other approaches, due to FCN employing VGG16 [19] as the feature extractor. VGG's superior classification capability translated to superior semantic segmentation, which was the reason that FCN is chosen as the generator for Pix2Pix architecture. Thus, the table 3.1 shows that the original Pix2Pix [1] network (UNet-Pix2Pix) which used UNet as the generator cannot reach the same performance as the FCN-Pix2Pix. The discriminator in the GAN architecture (FCN-

Pix2Pix) supervises generator outcomes from a global aspect. Thus, the proposed FCN-Pix2Pix model achieved robust and superior performance compared with all other state-of-the-art semantic segmentation neural networks considered, including FCN. Therefore, GAN architecture also improves performance over just including the generator.

Dataset: PST900 RGB Dataset					
Network	Extinguisher	Backpack	Drill	Survivor	mIoU
ERFNet	0.6118	0.6528	0.4240	0.4169	0.5263
MAVNet	0.2831	0.5850	0.3367	0.0901	0.3237
UNet	0.4928	0.6364	0.4026	0.2337	0.4413
Fast-SCNN	0.3454	0.6679	0.2063	0.2053	0.3562
PST-SegNet	**0.6814**	0.6990	0.5151	0.4989	0.5986
FCN	0.6133	0.7768	0.5345	0.6371	0.6404
UNet-Pix2Pix	0.5735	0.6903	0.4927	0.2599	0.5041
FCN-Pix2Pix	0.6036	**0.7872**	**0.5638**	**0.6572**	**0.6529**

Table 3.1: Performance metrics for the considered networks trained and tested using the PST900 training and testing datasets, respectively. The top five network experiments (ERFNet - PST-SegNet) on the table are done by [12], and the other three network experiments (FCN - FCN-Pix2Pix) are done by this work.

Chapter 4

Virtual Dataset from Simulation to Real

4.1 Introduction

The NCTU ARG Lab [6] used a real world robot arm pick-and-place dataset to train object detection [5]. However, data collection was time and human resource expensive, and training the models with virtual datasets produce poor performance for eventual real application. Therefore, this work proposes to build a GAN Sim2Real dataset to transform the virtual dataset from simulation to real.

This section discusses training SSD object detection [3] models using different datasets, and verifies the GAN Sim2Real dataset provides better performance than the virtual or other converted datasets.

4.2 Related Work

Several methods have been developed to transform image data from simulation to real with sufficient quality that the outcomes can be used as real-world data directly, i.e., models trained on the transformed datasets perform well when applied to actual real world datasets.

4.2.1 Histogram Matching

Traditional image processing approaches create virtual images for training by adding noise or adjusting image coloring. However, the virtual images are invariably much more "perfect" than the real images, i.e., the virtual images are less effected by noise, generally exclude blurring or unbalanced lighting effects, despite the camera is always moving between images, and lighting conditions change according to different locations and angles. Therefore, adding image blurring and salt-and-pepper noise can make "perfect" virtual images more imperfect, and more similar

to real-world images.

Histogram matching between real and virtual images (histogram specification) is an efficient and useful approach produce more realistic virtual images [26] [27] [28]. Image histograms represent the number of pixels for each tonal value in the image, and can describe the image hue distribution. We can calculate appropriate mapping functions between virtual and real image histograms, transforming the virtual image into a new image with matching histogram to real images.

Histogram matching and adding noise attempt to imitate real-world image aspects. However, these approaches are somewhat limited, and the range and domain they can successfully transform are insufficient. Therefore, deep neural network developments offer more reasonable solutions for simulation to real transformation.

4.2.2 Neural Style Transfer

Gatys et al. proposed the neural style transfer method [29], [30], which takes two images as input, for style and content, respectively. Both images are passed through a CNN (e.g. VGG19 [19]) for feature extraction. Style and content layers are defined manually by choosing certain layers from the CNN, and style and content loss are subsequently calculated according to the selected layers. The training process minimizes both style and content loss, and the network outputs image combining content and style from the corresponding input images.

4.2.3 Domain Randomization

In contrast with neural style transfer, domain rationalization adapts the input to the target domain [31], [32]. Object colors and textures are randomized in the simulator to ensure wide domain randomization dataset distribution, in principle covering the target domain we want to include.

Domain randomization is mainly used for grasping affordance or structural prediction from simulation to real. However, this approach is not suitable for this work since image color, pattern, and texture are all randomized, and these features are key points for object detection. Therefore, objects cannot be correctly detected and classified in the real world.

4.3 Method

4.3.1 NCTU-Brandname Dataset

In this work, I take NCTU-Brandname Dataset as the benchmark dataset. It is a 20-category object image dataset with pixel-wise label masks, the objects are mainly for robot arms pick and place mission, and all the image data are collected in the real environment and the labeled masks are labeled manually. The NCTU-Brandname Dataset is divided into training and testing datasets, each of them contains 25667 and 2851 image data with corresponding labeled masks, and it was published in [5] by NCTU ARG Lab [6].

4.3.2 Unity Virtual Dataset

The NCTU pick-and-place [5] environment included placing a red tote on a table with various objects inside in random poses. A camera above the table captured images of the whole scene. To ensure the virtual dataset is suitable to train systems that are subsequently applied in the real world, virtual and real environments should as similar as possible. Therefore, I chose the Unity [33] simulator, because it can not only can create a realistic 3D virtual environment but can also use an virtual camera to obtain various virtual scene images for a corresponding labeled image. The followings describe building the Unity virtual dataset:

Create Virtual Environment

First, 3D object models were created in CAD software (3D Builder), with textures imported from high resolution images of real objects. The resulting 3D virtual object model appeared quite realistic. A 3D red tote model was then built with similar specification and hue to the real tote. Real background images were pasted to the Unity background to make the virtual environment look very similar to the real world.

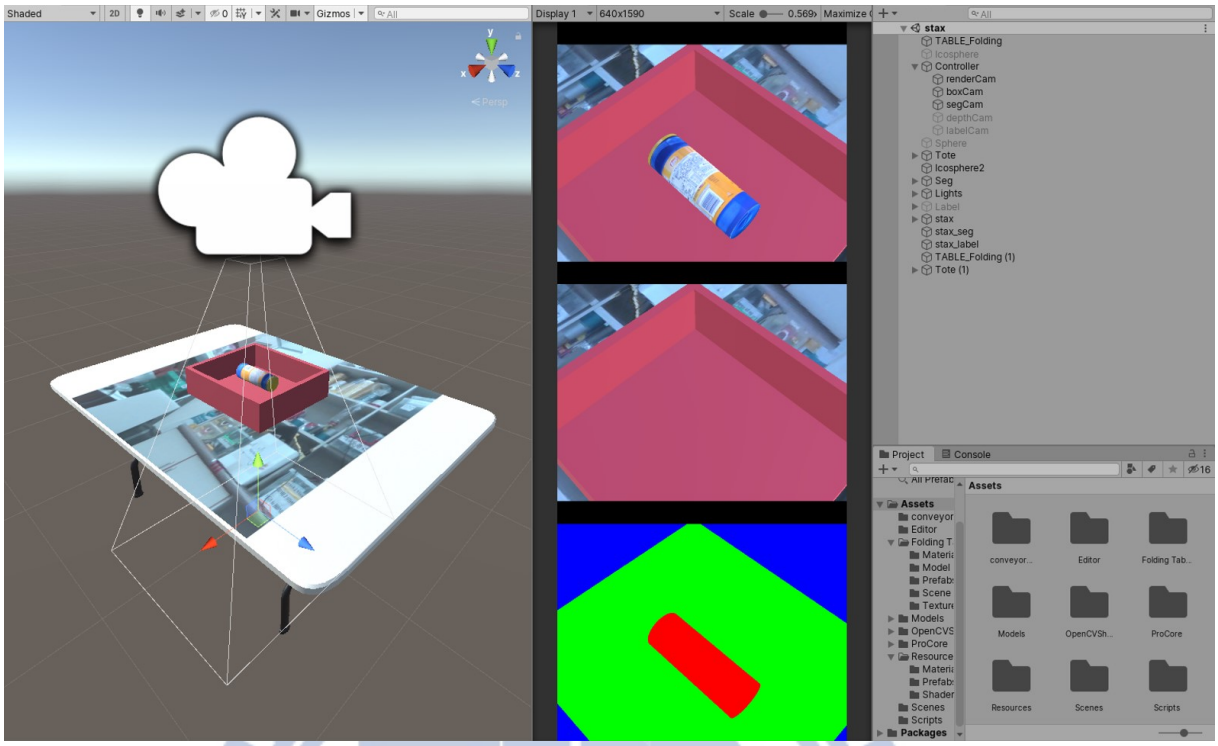


Figure 4.1: Data collection procedure for the Unity virtual environment: (left pane) 3D virtual environment created in Unity; (center pane) (top to bottom) Unity image, Unity box image, labeled mask.

Create Virtual Dataset

The basic virtual environment included 3D object models, 3D tote model and environment background, and we then created many different scenes to expand the Unity virtual dataset diversity. Fortunately, Unity enables programming in C# to create different object poses and camera angles, providing multiple virtual environment scenes. Unity can also automatically generate labeled images for the corresponding virtual scene image. Figure 4.1 shows the Unity virtual environment comprised a virtual camera capturing the scene, i.e., generating Unity images, and the corresponding labeled masks.

4.3.3 Real Tote Dataset

The Real Tote dataset was used as the target real-world domain for the simulation to real task, to allow training a deep CNN model or perform histogram specification to ensure realistic virtual images. For this particular case I took images of the actual red tote, which was similar to that used by NCTU [5], at different angles and distances. The final Red Tote dataset included

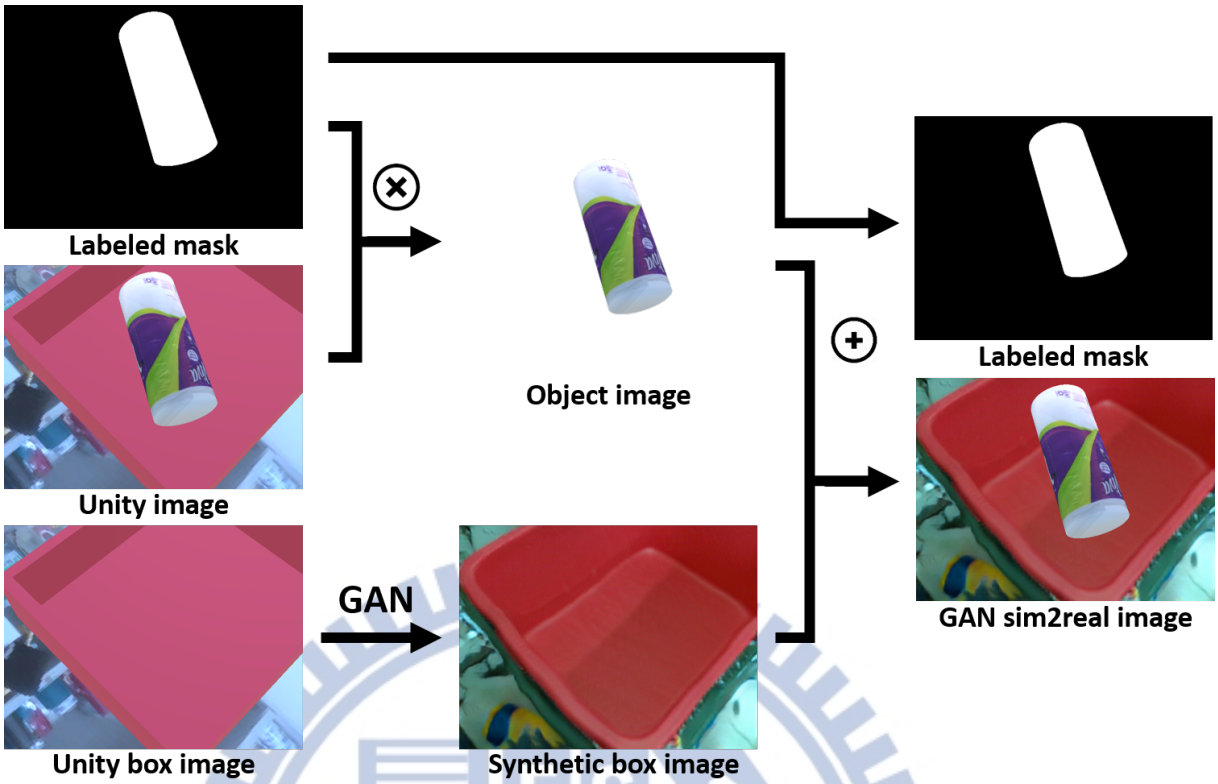


Figure 4.2: Architecture to generate the GAN Sim2Real dataset.

4752 tote images without objects inside Fig. 4.3.

Figure 4.3: First row is the data in Real Tote Dataset, and the second row is the Unity box images in Unity Virtual Dataset.

4.3.4 GAN Sim2real Dataset

I propose an SSIM-CycleGAN method to create the GAN Sim2Real (real) dataset from the Unity virtual (simulation) dataset, so the new dataset can be used in real world scenarios. Figure 4.2 shows the underlying architecture included SSIM-Cycle (based on CycleGAN [2]) to convert the simulation to real dataset. I only trained the SSIM-CycleGAN model to transform Unity box images rather than Unity images because we need extremely high accuracy labeled

masks for training. Hence if we convert the whole Unity image dataset to real images, we risk failing to match labeled masks with the newly generated image, and 3D object models in Unity itself were created from real images. Section 4.3.6 describes the GAN model architecture and the training details will also be described in the Sec. 4.3.6.

We first apply bitwise AND to the labeled mask and Unity images to generate object images that only contain the object itself. Unity box images are then passed through the SSIM-CycleGAN model to generate realistic looking synthetic box images. Finally, object images are added to the synthetic box images to create GAN sim2real images comprising GAN Sim2Real dataset along with the corresponding high accuracy labeled masks.

4.3.5 Histogram Matching Dataset

Object images were generated from the Unity dataset following the same process as for the GAN Sim2Real dataset. Then one image was randomly chosen from the Real Tote dataset and the Unity box image modified match histograms with the Real Tote image, generating a new matching box image. The new object image was added to the new matching box image to create a matched histogram image, and saved into the Matched Histogram dataset along with the corresponding labeled mask.

4.3.6 Network Architectures

Figure 4.6 shows that the original CycleGAN [2] approach causes structure distortion. The only constraint between generator input and output (Fig. 4.5) is the real or fake prediction from the discriminator, which would lead to generator overfitting. However, although cycle-consistency loss design can regularize the generator to avoid overfitting [2], if the data in domains A and B are not sufficiently diverse (as in this case), cycle-consistency loss will not prevent this problem. Therefore, I propose the SSIM-CycleGAN approach based on CycleGA [2], where structural similarity (SSIM) [34] [35] is a quality assessment based on structural information degradation. This takes two images as inputs and combines their comparative luminance,

Figure 4.4: Example images from (top to bottom) Unity virtual, Matched Histogram, GAN Sim2Real, and NCTU-Brandname datasets.

contrast, and structure to obtain an SSIM index Eq. 4.1.

$$SSIM(x,y) = [l(x,y)]^{\alpha}[c(x,y)]^{\beta}[s(x,y)]^{\gamma} \tag{4.1}$$

The SSIM loss in SSIM-CycleGAN calculates structure similarity differences between generator inputs and outputs, ensuring the generated image preserves the original image structure. Figure 4.6 shows that using CycleGAN leads to obvious structural distortion; whereas SSIM-CycleGAN, incorporating the SSIM loss constraint, can avoid structural distortion, producing more realistic (real-world) images.

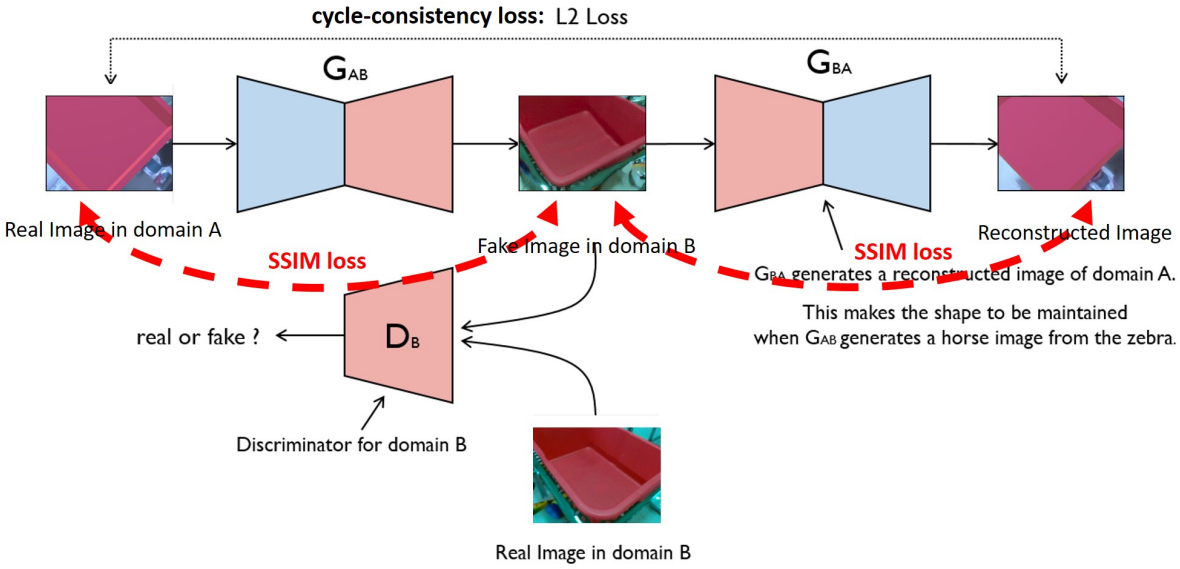


Figure 4.5: Proposed SSIM-CycleGAN network architecture based on CycleGAN [2] with additional structural similarity (SSIM) loss between generator input and output to avoiding structure distortion

4.4 Experiments

4.4.1 Experiment Design

We used the Unity virtual; Matched Histogram; GAN Sim2Real; and NCTU-Brandname (from NCTU ARG Lab [5] as benchmark) datasets for the experiments. In real-world logistic robot arm missions, the robot system should automatically detect object category and location, and then let the robot arm grasp the object. Therefore, the experimental datasets (including NCTU-Brandname) were trained on the SSD object detection network [3] and evaluated on the NCTU-Brandname testing dataset.

4.4.2 Evaluation Metric

I used the mean average precision (mAP) evaluation metric, a popular metric to measuring object detection accuracy, which computes average precision for recall = [0,1]. I needed the precision-recall (PR) curve before calculating average precision (AP). The PR curve is the recall

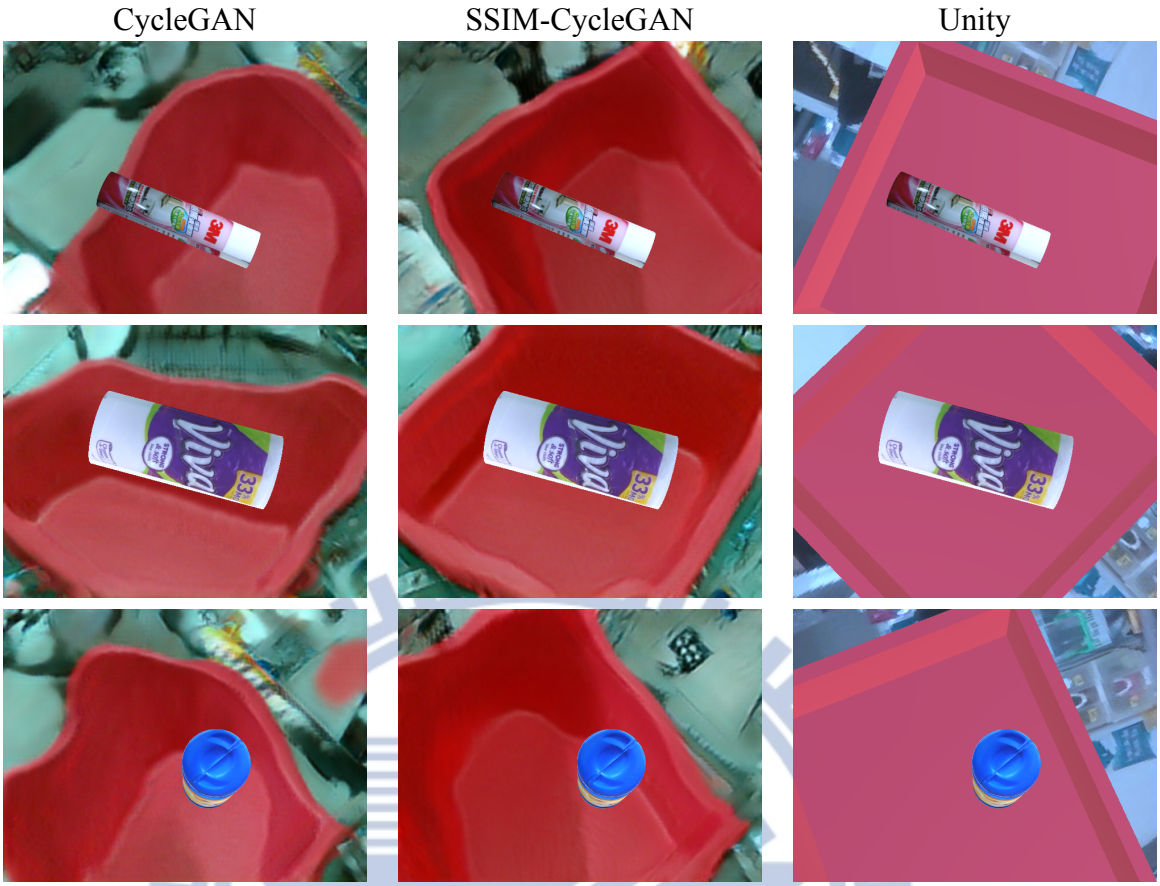


Figure 4.6: CycleGAN causes structural distortion, making the final image less similar to real images, however using SSIM-CycleGAn can avoid this problem due to the structure constrain by the SSIM loss.

and precision for a given IoU threshold for different predicted confidence levels,

$$Precision = \frac{TP}{TP + FP} \quad Recall = \frac{TP}{TP + FN} \tag{4.2}$$

where TP = true positive, TN = true negative FP = false positive, FN = false negative; AP is the area under the PR curve, and mAP as the numeric mean AP for all categories.

4.4.3 Results and Discussions

Table 4.1 shows the SSD model trained with the Unity virtual dataset achieved poor performance, which is only slightly improved by incorporating the Matched Histogram dataset. However, Training the SSD model with GAN Sim2Real dataset dramatically improved performance, with mAP @0.5 IoU increasing from 0.5 to 70.23 compared to using the Unity virtual

dataset, which strongly verifies the proposed SSIM-CycleGAN model as useful and powerful. The table also shows that using SSIM-CycleGAN is better than CycleGAN as the GAN architecture of the GAN Sim2Real Dataset.

Figure 4.4 shows that the SSD model trained with the Unity virtual dataset hardly detects any objects in the red tote, whereas training with GAN Sim2Real dataset detect included object very well. Figure 4.7 shows that the PR curve @0.5 IoU when trained using the Unity virtual dataset is oddly shaped, such that we cannot extract meaningful information, i.e., SSD model trained with Unity virtual dataset predictions are too poor to provide useful detection. In contrast, Fig. 4.8 shows that the PR curve @05 IoU when trained with the GAN Sim2Real dataset confirms low precision for high recall value and vice versa, which is a much more reasonable PR curve shape.

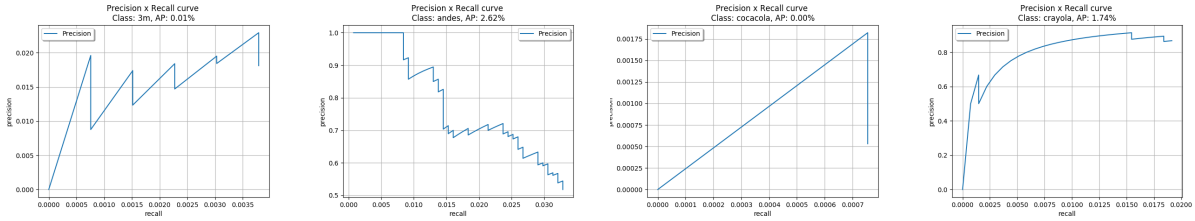
mAP	Unity Virtual	Histogram Matching	GAN Sim2Real (CycleGAN)	GAN Sim2Real (SSIM-CycleGAN)
@0.3IoU	2.28	9.35	67.78	71.41
@0.4IoU	0.86	5.77	67.11	70.90
@0.5IoU	0.5	3.78	66.35	70.23
@0.6IoU	0.4	2.66	64.85	68.97

Table 4.1: Performance metrics for SSD [3] on the NCTU-Brandname testing dataset after training with different transformed simulation to real datasets.

To show the GAN Sim2Real Dataset can really help us in the real world, I take the SSD model trained with GAN Sim2Real Dataset as the fine-tuned model M. Then I take 500 data from NCTU-Brandname training dataset, which contains 25667 data originally, to train on M. Later on, I compare it with the models trained only with 500 training data without any fine-tuning. Also, I train the whole NCTU-Brandname training dataset, and the result in table 4.2 shows that if we use model M as pre-trained model, we can use a small amount of real data (500 data) to do the fine-tuning, and the result can reach the high performance almost same as using great amount of real data (25667 data).

Number of NCTU-Brandname training data	500 data	500 data with Sim2Real pretrained model	25667 data (NCTU-Brandname training dataset)
@0.3IoU	60.35	94.75	96.93
@0.4IoU	60.28	94.61	96.79
@0.5IoU	60.15	94.35	96.58
@0.6IoU	59.74	93.92	96.21

Table 4.2: Using small amount of training data (500) to train them on SSD [3] with and without pre-trained model which is trained with GAN SimReal Dataset. The result shows that using pre-trained model trained with GAN Sim2Real Dataset can make the performance increase greatly, and almost have the same performance as using the whole NCTU-Brandname training dataset (25667 data) to train.

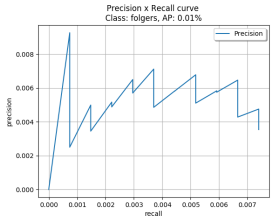


(a) 3m

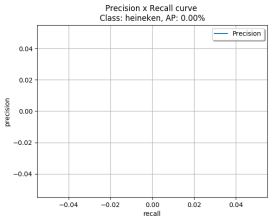
(b) Andes

(c) Cocacola

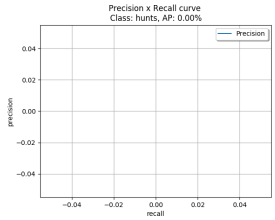
(d) Crayola



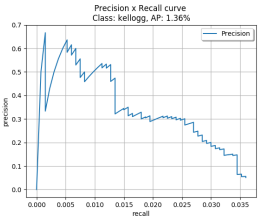
(e) Folgers



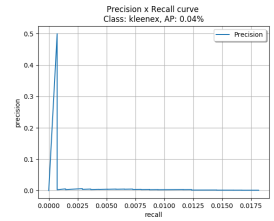
(f) Heineken



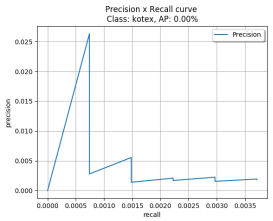
(g) Hunts



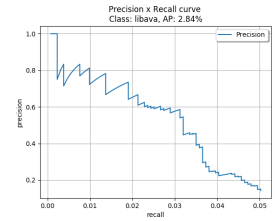
(h) Kellogg



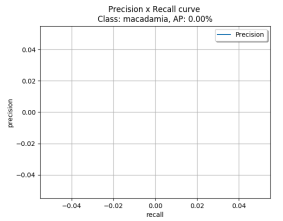
(i) Kleenex



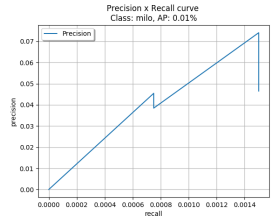
(j) Kotex



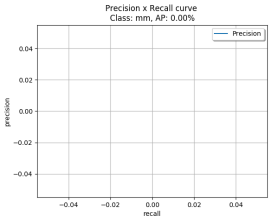
(k) Libava



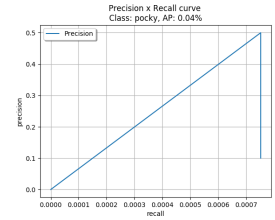
(l) Macadamia



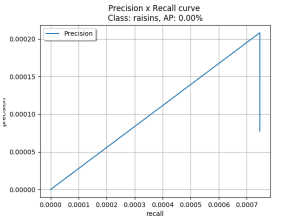
(m) Milo



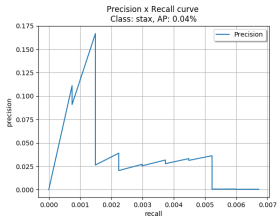
(n) mm



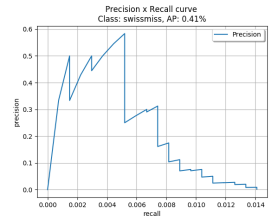
(o) Pocky



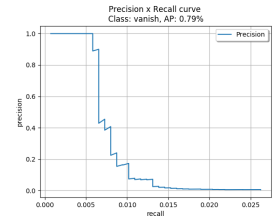
(p) Raisins



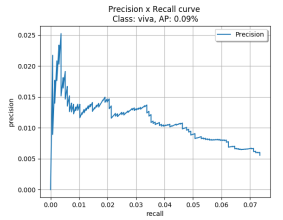
(q) Stax



(r) Swissmiss

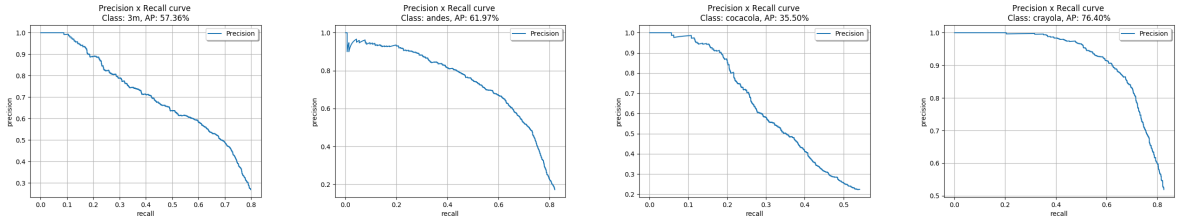


(s) Vanish

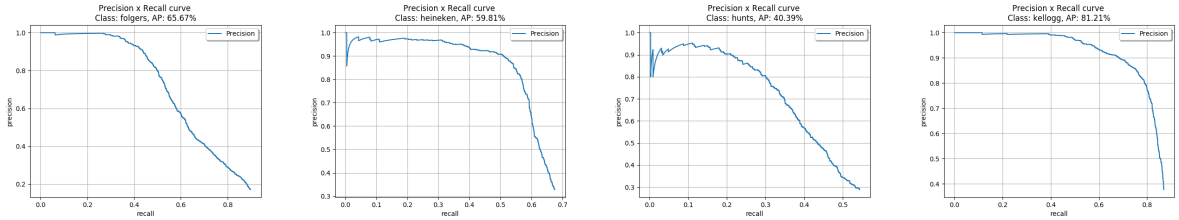


(t) Viva

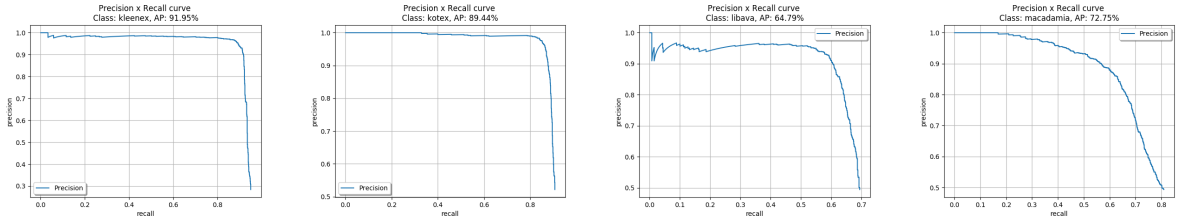
Figure 4.7: PR curves @0.5IoU for SSD models trained on the Unity virtual dataset



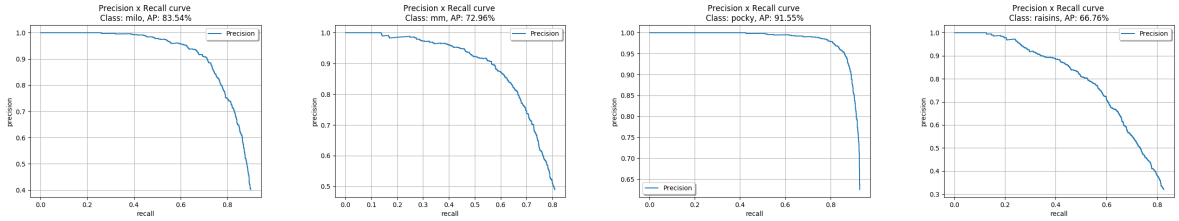
(a) 3m (b) Andes (c) Cocacola (d) Crayola



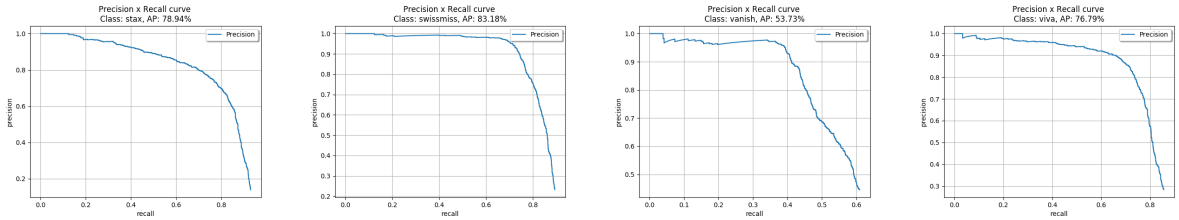
(e) Folgers (f) Heineken (g) Hunts (h) Kellogg



(i) Kleenex (j) Kotex (k) Libava (l) Macadamia

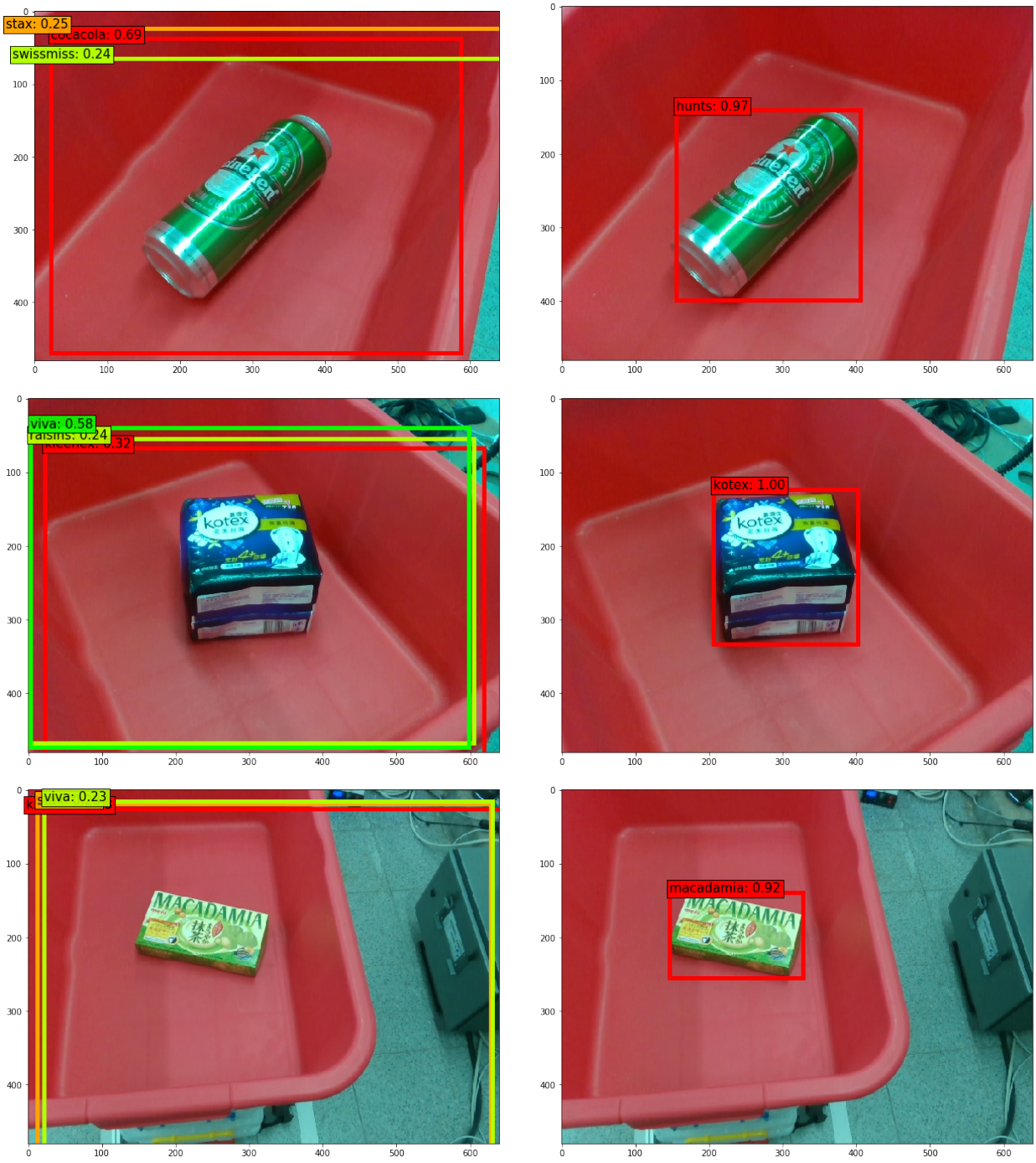


(m) Milo (n) mm (o) Pocky (p) Raisins



(q) Stax (r) Swissmiss (s) Vanish (t) Viva

Figure 4.8: PR curve @0.5IoU for SSD model trained on the GAN Sim2Real dataset



Unity virtual dataset GAN Sim2Real dataset

Figure 4.9: Prediction results for SSD models trained on Unity virtual and GAN Sim2Real datasets

Chapter 5

Conclusions and Future Works

I used GAN architectures to solve semantic segmentation and virtual dataset from simulation to real problems. Experiment results confirm that the proposed GAN based approaches achieved considerably superior performance than current state-of-the-art approaches.

Single generative models can create labeled mask images, and images with new styles, etc., but many drawbacks remain, including blurring, noise, and/or checkerboard artifacts. Extending the model to include a discriminate to supervise the generative model from a global aspect, provides considerably improved performance, which is also the main GAN concept.

Two experiments confirmed that the proposed GAN based models were suitable for general computer vision problems and could be implemented for real-robot missions.

GAN technology has already shown remarkable capability to accomplish tasks even humans cannot. The current outcomes confirm that GAN technology can have more applications in real-robot missions, solving real world application problems, and contributing to a more convenient future.

References

[1] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[2] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[3] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 21–37.

[4] DARPA. (2019) Darpa subterranean challenge. [Online]. Available: https://www.subtchallenge.com/

[5] S.-H. L. Yung-Shan Su, "Pose-aware placing with semantic labels - brandname-based affordance prediction and cooperative dual-arm active manipulation," 2019. [Online]. Available: https://arg-nctu.github.io/publications/text-pick-n-place-paper.pdf

[6] H.-C. Wang. (2016) Icn9005 robotic vision. [Online]. Available: https://arg-nctu.github.io/

[7] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 2672–2680. [Online]. Available: http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf

[8] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas, "Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks," in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[9] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 2672–2680. [Online]. Available: http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf

[10] G. Antipov, M. Baccouche, and J. Dugelay, "Face aging with conditional generative adversarial networks," in *2017 IEEE International Conference on Image Processing (ICIP)*, Sep. 2017, pp. 2089–2093.

[11] A. P. Aitken, C. Ledig, L. Theis, J. Caballero, Z. Wang, and W. Shi, "Checkerboard artifact free sub-pixel convolution: A note on sub-pixel convolution, resize convolution and convolution resize," *CoRR*, vol. abs/1707.02937, 2017. [Online]. Available: http://arxiv.org/abs/1707.02937

[12] S. S. Shivakumar, N. Rodrigues, A. Zhou, I. D. Miller, V. Kumar, and C. J. Taylor, "Pst900: Rgb-thermal calibration, dataset and segmentation network," *ArXiv*, vol. abs/1909.10980, 2019.

[13] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[14] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *CoRR*, vol. abs/1804.02767, 2018. [Online]. Available: http://arxiv.org/abs/1804.02767

[15] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[16] R. Girshick, "Fast r-cnn," in *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[17] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*

28, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 91–99. [Online]. Available: http://papers.nips.cc/paper/5638-faster-r-cnn-towards-real-time-object-detection-with-region-proposal-networks.pdf

[18] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[19] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv 1409.1556*, 09 2014.

[20] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[21] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds. Cham: Springer International Publishing, 2015, pp. 234–241.

[22] A. Odena, V. Dumoulin, and C. Olah, "Deconvolution and checkerboard artifacts," *Distill*, 2016. [Online]. Available: http://distill.pub/2016/deconv-checkerboard

[23] E. Romera, J. M. Álvarez, L. M. Bergasa, and R. Arroyo, "Erfnet: Efficient residual factorized convnet for real-time semantic segmentation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, pp. 263–272, Jan 2018.

[24] T. Nguyen, T. Özaslan, I. D. Miller, J. F. Keller, S. S. Shivakumar, G. Loianno, C. J. Taylor, V. Kumar, J. H. Harwood, and J. M. Wozencraft, "Mavnet: an effective semantic segmentation micro-network for mav-based tasks," *CoRR*, vol. abs/1904.01795, 2019. [Online]. Available: http://arxiv.org/abs/1904.01795

[25] R. P. K. Poudel, S. Liwicki, and R. Cipolla, "Fast-scnn: Fast semantic segmentation network," *CoRR*, vol. abs/1902.04502, 2019. [Online]. Available: http://arxiv.org/abs/1902.04502

[26] D. Coltuc, P. Bolon, and J. . Chassery, "Exact histogram specification," *IEEE Transactions on Image Processing*, vol. 15, no. 5, pp. 1143–1152, May 2006.

[27] G. Thomas, D. Flores-Tapia, and S. Pistorius, "Histogram specification: A fast and flexible method to process digital images," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 5, pp. 1565–1578, May 2011.

[28] K. Inoue, H. Kenji, and K. Urahama, "Rgb color cube-based histogram specification for hue-preserving color image enhancement," *Journal of Imaging*, vol. 3, no. 3, 9 2017.

[29] L. A. Gatys, A. S. Ecker, and M. Bethge, "A neural algorithm of artistic style," *CoRR*, vol. abs/1508.06576, 2015. [Online]. Available: http://arxiv.org/abs/1508.06576

[30] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 2414–2423.

[31] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2017, pp. 23–30.

[32] S. James, P. Wohlhart, M. Kalakrishnan, D. Kalashnikov, A. Irpan, J. Ibarz, S. Levine, R. Hadsell, and K. Bousmalis, "Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[33] Unity. (2019) Unity real-time development platform. [Online]. Available: https://unity.com/

[34] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, April 2004.

[35] G. Chen, C. Yang, and S. Xie, “Gradient-based structural similarity for image quality assessment,” in *2006 International Conference on Image Processing*, Oct 2006, pp. 2929–2932.