An Analysis of DualGAN: Unsupervised Dual Learning for Image-to-Image Translation

Nitin Nataraj
Department of Computer Science
University at Buffalo
Buffalo, NY 14214
nitinnat@buffalo.edu

Karan Hora
Department of Computer Science
University at Buffalo
Buffalo, NY 14214
karanhor@buffalo.edu

Priyanshi Shukla
Department of Computer Science
University at Buffalo
Buffalo, NY 14214
pshukla3@buffalo.edu

Abstract

In this paper, we apply unsupervised dual learning model via DualGANs, to perform the task of image-to-image translation. The approach has been inspired by dual learning from natural language translation. The architecture has two generative adversarial networks namely Primal and Dual GANs. Primal GAN which learns to translate images from one domain to the other and the Dual GAN that learns to invert the task. The objective is to discriminate between the generated fake sample from the real ones. Experiments have been conducted on multiple unlabeled datasets such as the Photo-sketch dataset, oil-chinese paintings, materials dataset, label map-facades, maps-aerial photo and day-night translation. Also, the scope of this mechanism has been extended to Multi GAN implementation and image segmentation task as well.

1 Introduction

Computer Vision and Image processing tasks such as image segmentation, stylization and abstraction posed as image – image translation problems have been tackled with supervised approach like FCNs. However, acquiring labelled training data for the same is highly time consuming and not feasible. Therefore, an unsupervised approach of DualGANs has been implemented. It is inspired by dual learning from natural language translation, which enables translators to be trained from two sets of unlabeled images from two domains. The GAN discriminators are trained adversarially with the translators to capture domain distributions. Fully connected convolutional networks which accommodate 2d structure of images of naturally, are used as translators. 2 sets of unlabeled images characterizing each domain is given as input to the GANs. Thus, the DualGAN simultaneously learns two reliable image translators from one domain to the other and hence can operate on a wide variety of image-to-image translation tasks.

2 Related Work

Some of the related previous works on such unsupervised learning techniques has been reviewed. Ian Goodfellow et al [1], uses cGANs for image generation on close lebels, attributes texts etc. Most of these image-conditional models were developed for specific applications and not specific ones. The

general purpose solution described by Isola et al [2] requires significant amount of labelled image pairs The unsupervised mechanism for cross-domain image conversion presented by Taigman et al. [3] can train an image-conditional generator without paired images, but relies on a sophisticated pre-trained function that maps images from either domain to an intermediate representation, which requires labeled data in other formats. Dual learning was first proposed by Xia et al. [4] to reduce the requirement on labeled data in training English-to-French and French-to-English translators. The French-to-English translation is the dual task to English-to-French translation, and they can be trained side-by-side. The key idea of dual learning is to set up a dual-learning game which involves two agents, each of whom only understands one language, and can evaluate how likely the translated are natural sentences in targeted language and to what extent the reconstructed are consistent with the original. In CycleGAN, a concurrent work by Zhu et al. [5], the same idea for unpaired image-to-image translation is proposed, where the primal-dual relation in DualGAN is referred to as a cyclic mapping and their cycle consistency loss is essentially the same as DualGAN reconstruction loss. Superiority of CycleGAN has been demonstrated on several tasks where paired training data hardly exist, e.g., in object transfiguration and painting style and season transfer.

3 Network Architecture

Given two sets of unlabeled and unpaired images sampled from domains U and V, respectively, the primal task of DualGAN is to learn a generator $G_{\mathrm{A}} : U \rightarrow V$ that maps an image u belonging to U to an image v belonging to V , while the dual task is to train an inverse generator $G_{\mathrm{B}} : VU$. To realize this, we employ two GANs, the primal GAN and the dual GAN. The primal GAN learns the generator G_{A} and a discriminator $\mathrm{D_A}$. that discriminates between G_{A}'s fake outputs and real members of domain V Analogously, the dual GAN learns the generator G_{B} and a discriminator D_{B}. The overall architecture and data flow are illustrated in Fig. 1.

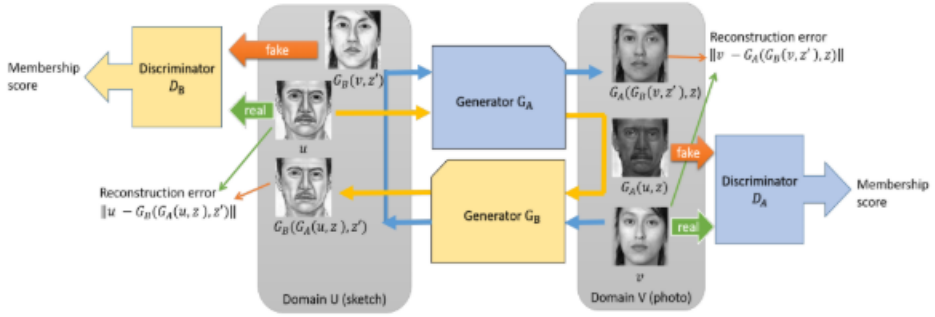


Figure 1: Architecture of the DualGAN

Image $u \in U$ is translated to domain V using G_{A}. How well the translation $G_{\mathrm{A}}(u, z)$ fits in V is evaluated by D_{A}., where z is random noise, and so is z'. $G_{\mathrm{A}}(u, z)$ is then translated back to domain U using G_{B}, which outputs $G_{\mathrm{B}}(G_{\mathrm{A}}(u, z), z')$ as the reconstructed version of u. Similarly, $v \in V$ is translated to U as $G_{\mathrm{B}}(v, z')$ and then reconstructed as $G_{\mathrm{A}}(G_{\mathrm{B}}(v, z'), z)$. The discriminator D_{A}. is trained with v as positive samples and $G_{\mathrm{A}}(u, z)$ as negative examples, whereas D_{B} takes u as positive and $G_{\mathrm{B}}(v, z')$ as negative. Generators G_{A} and G_{B} are optimized to emulate "fake" outputs to blind the corresponding discriminators D_{A}. and DB, as well as to minimize the two reconstruction losses $||G_{\mathrm{A}}(G_{\mathrm{B}}(v, z'), z) - v||$ and $||G_{\mathrm{B}}(G_{\mathrm{A}}(u, z), z') - u||$. The objective of discriminators is to discriminate the generated fake samples from the real ones. Here we use the loss format advocated by Wasserstein GAN (WGAN) [6]. The corresponding loss functions used in D_{A}. and D_{B}.are defined as:

$$l^{\mathrm{d}}{}_{\mathrm{A}}(u, v) = D_{\mathrm{A}}(G_{\mathrm{A}}(u, z)) - D_{\mathrm{A}}(v) \quad (1)$$
$$l^{\mathrm{d}}{}_{\mathrm{B}}(u, v) = D_{\mathrm{B}}(G_{\mathrm{B}}(u, z)) - D_{\mathrm{B}}(v) \quad (2)$$
where $v \in V$ and $u \in U$

The same loss function is used for both generators G_{A} and G_{B} as they share the same objective. L1 distance is used to measure the recovery error, which is added to the GAN objective to force the translated samples to obey the domain distribution:

$$l^{\mathrm{g}}_{\mathrm{A,B}}(u,v) = \\ \lambda_{\mathrm{U}}||u - G_{\mathrm{B}}(G_{\mathrm{A}}(u,z),z')|| + \lambda_{\mathrm{V}}||v - G_{\mathrm{A}}(G_{\mathrm{B}}(v,z'),z)|| - D_{\mathrm{A}}(G_{\mathrm{B}}(v,z')) - D_{\mathrm{B}}(G_{\mathrm{A}}(u,z)) \quad (3)$$

Where u belongs to domain U and v belongs to domain V and λ_{U} , λ_{V} are two constant parameters. Depending on the application, λ_{U} and λ_{V} are typically set to a value within [100.0, 1, 000.0].

4 Network Configuration

The generators are configured with equal number of downsampling (pooling) and upsampling layers. And with skip connections between mirrored downsampling and upsampling layers as in [7], making it a U-shaped net. To allow low-level information to be shared between input and output. Without the skip layers, information from all levels must pass through the bottleneck causing significant loss of high- frequency information. The noise vectors z and z' mentioned in the network architecture were applied in the form of dropout to several layers of generators, during the train and test phase. The discriminators were constructed using Markovian patch GAN [8]. i.e. pixels beyond a specific patch are assumed to be independent of each other. Thus, the images are modelled at patch level which results in use of fewer parameters and faster computation. Here, the patch size is taken as 70X70 and image resolution is fixed at 256X256. The discriminator is run convolutionally across the image, averaging all responses to provide the ultimate output.

5 Training

Minibatch stochastic gradient descent approach with applied RMSProp solver is implemented for the network training [6]. The discriminators are trained for n_{critic} steps and only then train one step on generators. At each step the generators are not trained until discriminators have been trained. This is to ensure more reliable gradient information from the discriminators and to avoid the issue local saturation and vanishing gradients. The number of critic iterations per generator iteration i.e. ncritic is typically set to 2-4 and batch size is assigned as 1-4. The clipping parameter c is normally set in [0.01, 0.1]. The algorithm can be stated as follows:

Algorithm 1 DualGAN training procedure

Require: Image set U, image set V, GAN A with generator parameters θ_A and discriminator parameters ω_A, GAN B with generator parameters θ_B and discriminator parameters ω_B, clipping parameter c, batch size m, and n_{critic}

Randomly initialize ω_i, θ_i, $i \in \{A, B\}$
repeat
  for $t = 1, \ldots, n_{critic}$ **do**
    sample images $\{u^{(k)}\}_{k=1}^{m} \subseteq U$, $\{v^{(k)}\}_{k=1}^{m} \subseteq V$
    update ω_A to minimize $\frac{1}{m}\sum_{k=1}^{m} l_A^d(u^{(k)}, v^{(k)})$
    update ω_B to minimize $\frac{1}{m}\sum_{k=1}^{m} l_B^d(u^{(k)}, v^{(k)})$
    $clip(\omega_A, -c, c)$, $clip(\omega_B, -c, c)$
  end for
  sample images $\{u^{(k)}\}_{k=1}^{m} \subseteq U$, $\{v^{(k)}\}_{k=1}^{m} \subseteq V$
  update θ_A, θ_B to minimize $\frac{1}{m}\sum_{k=1}^{m} l^g(u^{(k)}, v^{(k)})$
until convergence

Figure 2: The DualGAN algorithm.

6 Results

The effectiveness of DualGAN was shown in terms of qualitative evaluation (Figures 2, 3, 4) and quantitative evaluation. And comparisons were made with simple GAN architecture and cGAN

architecture. We compare DualGAN with GAN and cGAN [2] on the following translation tasks: face photo ↔ sketch (Figure 2, 3), and OilChinese (Figure 4). In all these tasks, cGAN was trained with labeled (i.e., paired) data, whereas DualGAN and GAN were trained in unsupervised way. All three models were trained on the same training datasets and tested on novel data that does not overlap those for training. Compared to GAN, in almost all cases, DualGAN produced results that are less blurry, contain fewer artifacts, and better preserve content structures in the inputs and capture features (e.g., texture, color, and/or style) of the target domain. The improvements are attributed to the reconstruction loss, which forces the inputs to be reconstructable from outputs through the dual generator and strengthens feedback signals that encodes the targeted distribution. In many cases, DualGAN also compares favorably over the supervised cGAN in terms of sharpness of the outputs and faithfulness to the input images; Refer figures 2, 3, 4.

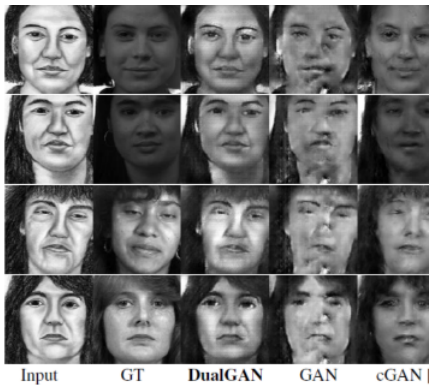


Figure 3: Photo ↔ sketch translation for faces. Results of DualGAN are generally sharper than those from cGAN, even though the former was trained using unpaired data, whereas the latter makes use of image correspondence

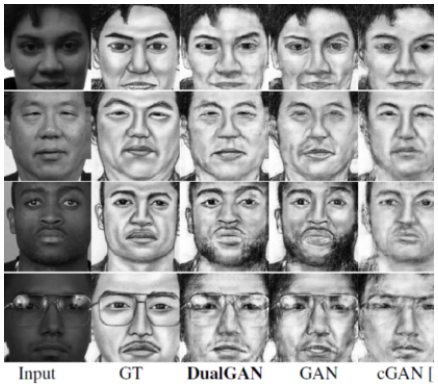


Figure 4: Results for sketch ↔ photo translation of faces. More artifacts and blurriness are showing up in results generated by GAN and cGAN than DualGAN.

The results we obtained on the sketch ↔ photo dataset using the DualGAN are shown in Figure 5. The results obtained match the original implementation and took a little over 8 hours to run on an Nvidia 1060 GTX GPU. The results obtained for both translations are crisp and appear very realistic.

Figure 5: Results obtained by our execution of the DualGAN implementation on the sketch $\leftrightarrow$ photo dataset. The network was trained for 50 epochs with a batch size = 1 and $\lambda_U = 20.0$ and $\lambda_V = 20.0$.

The authors also run their model on the materials dataset, a set of images of objects from various material domains such as plastic, wood and metal. The results on this dataset are shown in Figure 6.

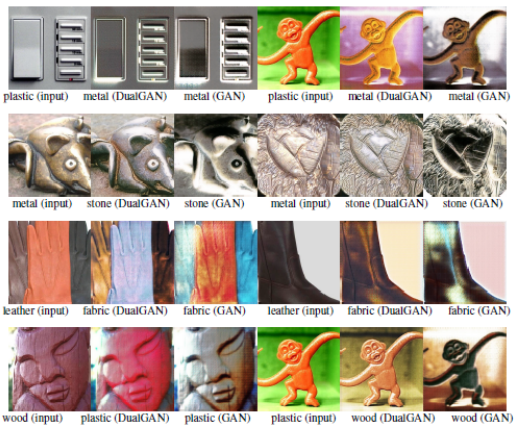


Figure 6: Experimental results for various material transfer tasks. From top to bottom, plastic$\leftrightarrow$metal, metal $\leftrightarrow$ stone, leather $\leftrightarrow$ fabric and plastic $\leftrightarrow$ wood.

For Quantitative evaluation, the authors of the DualGAN paper set up two user studies through Amazon Mechanical Turk (AMT). The "material perceptual" test evaluates the material transfer results, in which the authors mixed the outputs from all material transfer tasks (Figure 6) and let the Turkers choose the best match based on which material they believe the objects in the image are made of. For a total of 176 output images, each was evaluated by ten Turkers. An output image is rated as a success if at least three Turkers selected the target material type. Success rates of various material transfer results using different approaches are summarized in Figure 7, showing that DualGAN outperforms GAN by a large margin.

Task	DualGAN	GAN
plastic→wood	**2**/11	0/11
wood→plastic	**1**/11	0/11
metal→stone	**2**/11	0/11
stone→metal	**2**/11	0/11
leather→fabric	**3**/11	2/11
fabric→leather	**2**/11	1/11
plastic→metal	**7**/11	3/11
metal→plastic	**1**/11	0/11

Figure 7: Success rates of various material transfer tasks based on the AMT "material perceptual" test. There are 11 images in each set of transfer result, with noticeable improvements of DualGAN over GAN.

7 Possible Applications and Improvements

7.1 MultiGAN

We decided to extend the idea of DualGAN to that of MultiGAN, in which there are hypothetically n generators and n discriminators, corresponding to n image domains. A MultiGAN can be thought of as a clique of n nodes, with each node possessing both a generator and a discriminator, and representing a particular domain of images. For the purposes of our experiments, we have chosen to work with three GANs. The architecture of the MultiGAN with three GANs is similar to that of the DualGAN and is shown in Figure 8.

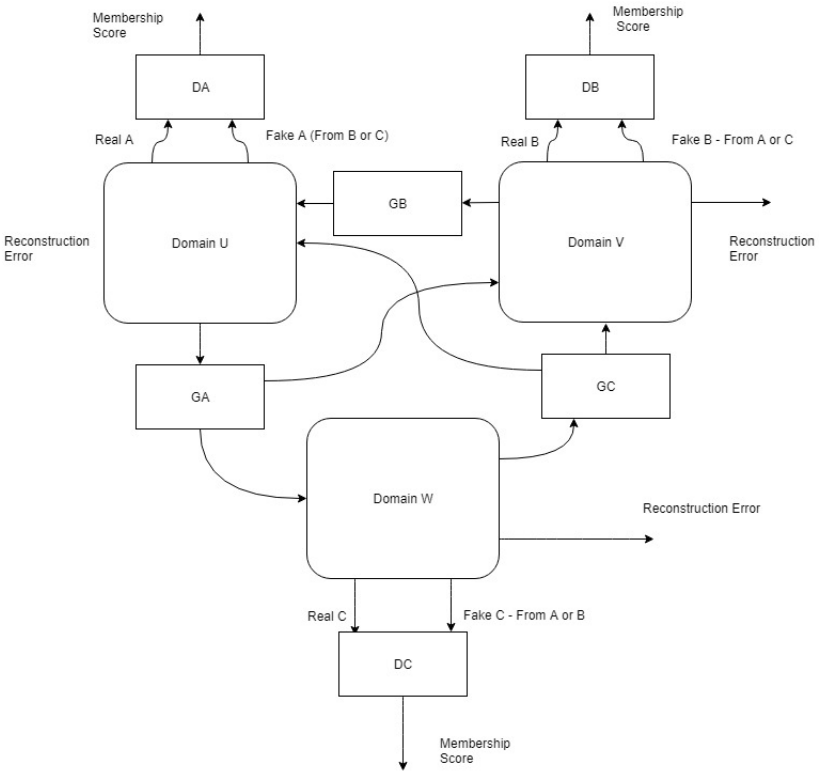


Figure 8: The MultiGAN architecture for the case with three GANs. Each GAN has it's own generator and discriminator, and pairs of GANs are trained alternately during the training procedure.

The loss functions for each GAN within the network are inspired from the DualGAN objective, and are shown in below.

$$l^{\mathrm{d}}{}_{\mathrm{A}}(u, v) = D_{\mathrm{A}}(G_{\mathrm{A}}(u, z)) - D_{\mathrm{A}}(v)$$

$$l^{\mathrm{d}}{}_{\mathrm{B}}(u, v) = D_{\mathrm{B}}(G_{\mathrm{B}}(u, z)) - D_{\mathrm{B}}(v)$$

$$l^{\mathrm{d}}{}_{\mathrm{C}}(u, v) = D_{\mathrm{C}}(G_{\mathrm{C}}(u, z)) - D_{\mathrm{C}}(v)$$

$$l^{\mathrm{g}}{}_{\mathrm{A,B}}(u, v) = \lambda_{\mathrm{U}}||u - G_{\mathrm{B}}(G_{\mathrm{A}}(u, z), z')|| + \lambda_{\mathrm{V}}||v - G_{\mathrm{A}}(G_{\mathrm{B}}(v, z'), z)|| - D_{\mathrm{A}}(G_{\mathrm{B}}(v, z')) - D_{\mathrm{B}}(G_{\mathrm{A}}(u, z))$$

$$l^{\mathrm{g}}{}_{\mathrm{A,C}}(u, v) = \lambda_{\mathrm{U}}||u - G_{\mathrm{A}}(G_{\mathrm{C}}(u, z), z')|| + \lambda_{\mathrm{V}}||v - G_{\mathrm{C}}(G_{\mathrm{A}}(v, z'), z)|| - D_{\mathrm{C}}(G_{\mathrm{A}}(v, z')) - D_{\mathrm{A}}(G_{\mathrm{C}}(u, z))$$

$$l^{\mathrm{g}}{}_{\mathrm{B,C}}(u, v) = \lambda_{\mathrm{U}}||u - G_{\mathrm{B}}(G_{\mathrm{C}}(u, z), z')|| + \lambda_{\mathrm{V}}||v - G_{\mathrm{C}}(G_{\mathrm{B}}(v, z'), z)|| - D_{\mathrm{C}}(G_{\mathrm{B}}(v, z')) - D_{\mathrm{B}}(G_{\mathrm{C}}(u, z))$$

During the training procedure, we take different pairs of GANs in an order-independent manner and minimize the respective loss functions for that pair. The parameters of a particular generator are updated $n - 1$ times, and a discriminator is updated $2(n - 1)$ times during the one training iteration. The total number of loss function components in terms of the number of GANs n can be given by ${}^{\mathrm{n}}2C_2$.

7.1.1 Results

The results of this architecture were obtained on the materials dataset, where we selected metal, wood and plastic to be the three domains for our 3-GAN architecture. The respective loss functions of all the networks are shown in Figure 9.

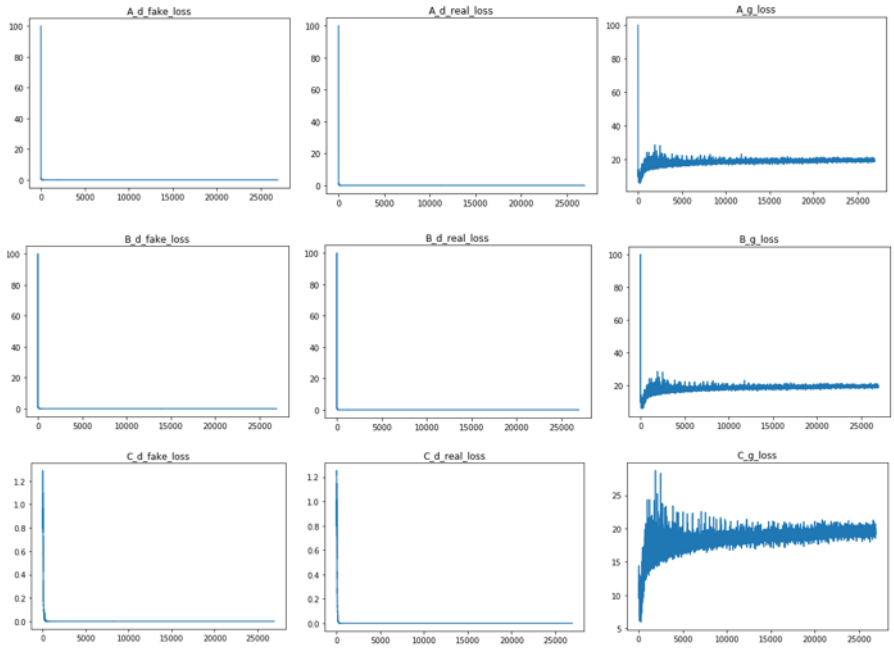
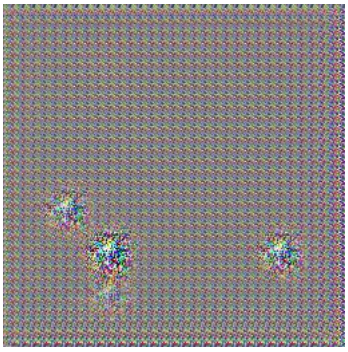


Figure 9: Discriminator and Generator losses plotted against number of seconds taken on an Nvidia 1060 GTX 6GB GPU and a 16GB RAM CPU for the 3-GAN. All loss functions decrease smoothly over time.



(a) Generator B trying to generate an image of domain C.

(b) Generator C generating an image of domain B from the fake image B.

Figure 10: Image results from the three-GAN.

As is well known in the GAN-community, training GANs is a daunting and unstable task. So the thought of obtaining an equilibrium by simultaneously minimizing six loss functions seems too far-fetched. The images obtained after running the trained architecture on the test images were not good. The GANs were unable to train properly and grids of grey cells with random splashes of color were obtained. Two such images are shown in Figure 10.
This failure to produce proper output could be attributed to the fact that the loss functions are updated with different domain pairs in every iteration, which possibly causes divergence in the training. A future direction could include careful selection of loss functions to be updated at each iteration, and possibly keeping the parameters of one GAN constant after it has already been trained in that iteration.

7.2 Image Segmentation

The DualGAN can be used to perform image segmentation. They produce sharp images when trained with high number of epochs. But for certain tasks which involve semantics-based labels they do not perform well. This is because of a lack of pixel and label correspondence information, which cannot be inferred from the training images. A semi-supervised approach or a warm start, using a small number of labeled data, can be explored as a potential solution.

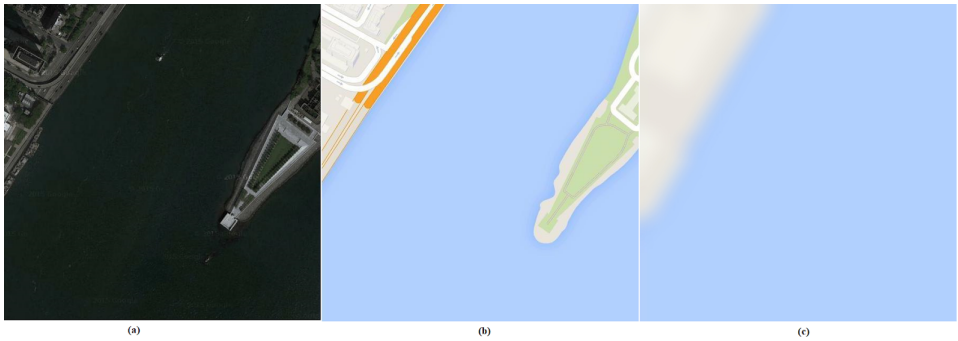


Figure 11: Aerial photo to map translation. (a) is the aerial photo and (b) is the actual map. The model on training after 20 epochs give result (c).The DualGan segments the image and also has to infer what colors correspond to the segments i.e. blue (water) and orange (highway)

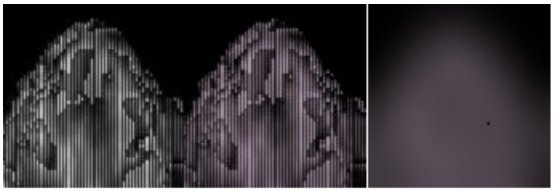


Figure 12: (a) A brain scan (b) Segmented brain scan (c) Output after 25 epoch training

8 Acknowledgements

We are humbly grateful to Dr. Sargur Srihari for being a great source of inspiration and for providing us the opportunity and the guidance as we worked on this project. We would also like to thank our TAs, Mihir Chauhan and Mohammad Shaikh for their continued support.

References

[1] 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, pages 2672–2680, 2014. 1, 2, 3, 5.

[2] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image -to-image translation with conditional adversarial networks. arXiv preprint arXiv:1611.07004, 2016. 1, 2, 3, 4, 5, 6, 7, 8.

[3] Y. Taigman, A. Polyak, and L. Wolf. Unsupervised cross-domain image generation. arXiv preprint arXiv:1611.02200, 2016. 1, 2

[4] Y. Xia, D. He, T. Qin, L. Wang, N. Yu, T.-Y. Liu, and W.-Y. Ma. Dual learning for machine translation. arXiv preprint arXiv:1611.00179, 2016. 1, 2, 3, 4.

[5] J. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In International Conference on Computer Vision (ICCV), to appear, 2017. 2, 7

[6] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. arXiv preprint arXiv:1701.07875, 2017. 3, 4

[7] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical Image Computing and Computer-Assisted Intervention, pages 234–241. Springer, 2015. 3

[8] C. Li and M. Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In European Conference on Computer Vision (ECCV), pages 702–716. Springer,2016. 1, 2, 3