TELECOM Paris

IP PARIS

Computer Vision

Domain adaptation: MNIST to SVHN

MS IA
Yuchen Xia
Salimatou Traore

Contents

1 Introduction

1.1 Context

Transfer learning and domain adaptation are critical branches of machine learning aimed at enhancing the generalization of models trained on a specific domain (source) to perform effectively on a new domain (target) where the data may significantly differ. This approach is especially valuable in situations where labeled data in the target domain are scarce or completely absent.

1.2 Problem Statement

In our case, the challenge is to transfer the knowledge from a model trained on the MNIST dataset, which consists of black and white handwritten digit images, to accurately classify images from the SVHN dataset, which contains color images of digits from house numbers.

1.3 Objective

The primary objective of this project is to develop a system capable of effectively classifying images from the SVHN dataset using only the knowledge and labels from the MNIST dataset, without access to SVHN labels during training. Given the distinct differences between these two datasets—notably, that they are unpaired and belong to separate domains—we have chosen to implement a CycleGAN approach. CycleGAN allows for domain adaptation through unpaired data by learning to map relationships between two distinct domains. Its GAN-based architecture is particularly adept at capturing and learning the distinguishing features and distributions of each domain. Through this process, the CycleGAN is trained to generate images with features characteristic of the target domain, thereby enabling models trained on transformed images to achieve high performance.
The effectiveness of our model is quantified through its classification accuracy on the SVHN dataset. We aim to optimize this accuracy by experimenting with various hyperparameters, including the number of training iterations, learning rates, and optimizer choices.

Figure 1: Two digit datasets: MNIST (left) and SVHN (right)

2 Methodology

2.1 CycleGAN

2.1.1 Architecture and Training

The structure of CycleGAN consists of two generators ($G_{\mathbf{MNIST \to SVHN}}$ and $G_{\mathbf{SVHN \to MNIST}}$) and two discriminators (D_{MNIST} and D_{SVHN}). The generators are responsible for converting the images from one domain to another, and the discriminators are responsible for determining the authenticity of the generated SVHN-style images and MNIST-style images.
These generators are constructed based on the ResNet architecture. In generators, the image styles are converted by encoding and decoding the images. The structure of ResNet is usually more stable and easier to train than regular deep networks. Using ResNet as a generator improves the stability and convergence of the model and speeds up the training process.
The discriminators applied a PatchGAN construction, it divides the image into local regions and performs independent discriminations for each local region, which allows the discriminator to better capture local details and features of the image without considering the global features of the entire image. PatchGAN requires fewer parameters, and is therefore more efficient and faster to train.

The training employed unpaired MNIST and SVHN images, leveraging adversarial and cycle consistency losses to refine image translation and preserve original image integrity. We set the learning rate at 0.0002, used the Adam optimizer, and ran the process for 50 epochs. To overcome challenges like mode collapse and training instabilities, we meticulously adjusted the model's architecture and parameters.

2.1.2 Evaluation

To ensure that the generated new images closely resemble the original ones, we need to minimize the loss of the CycleGAN. This loss consists of two parts: GAN loss($Loss_{GAN}$) and cycle loss($Loss_{cycle}$). The GAN loss, calculated using Mean Squared Error, measures the difference between the generated images and the real images from the target domain. The cycle loss, calculated using L1 Loss, quantifies the absolute difference between the images after undergoing cycle transformation and the original images, ensuring consistency when the images cycle back from one domain to another.
For example, for an image x in dataset MNIST and its SVHN's corresponded image y, the calculation of GAN loss is $Loss_{GAN} = \frac{1}{N}\sum_{i=1}^{N}(G_{\text{MNIST}\to\text{SVHN}}(x) - y)^2$ where N is the number of pixels. And the cycle loss is $Loss_{cycle} = \frac{1}{N}\sum_{i=1}^{N}(G_{\text{MNIST}\to\text{SVHN}}|G_{\text{SVHN}\to\text{MNIST}}(x)) - x|$ where N is the number of pixels.

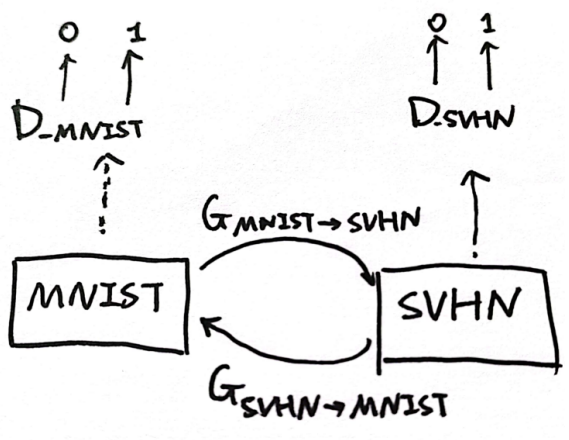

Figure 2: The structure of CycleGAN

2.2 LeNet Classifier

2.2.1 Architecture

The LeNet classifier used in this study is a convolutional neural network comprising two convolutional layers followed by max-pooling layers which is aimed to reduce the size of the feature map while retaining the most important features, two fully connected layers, two 2D Dropout layers used to prevent overfitting, randomly sets some of the input neurons to zero. and a final output layer that classifies the images into ten categories(0 - 9). This architecture is well-suited for image classification tasks due to its efficiency in learning spatial hierarchies in image data.

2.2.2 Integration with CycleGAN

The images generated by $G_{\text{MNIST}\to\text{SVHN}}$ were used as training data for the LeNet classifier. This approach leverages the transformed characteristics of SVHN-like images while retaining the labels from the original MNIST dataset, thus enabling effective domain adaptation.

2.2.3 Training the Classifier

The classifier was trained exclusively on the MNIST-like images transformed to resemble the SVHN dataset. The training involved using the cross-entropy loss function to measure the discrepancy between the predicted labels and the true MNIST labels, optimizing the network using the Adamax optimizer with a learning rate of 0.0002. The classifier training was performed for 10 epochs, and early stopping was employed to prevent overfitting, based on the validation loss.

Figure 3: The structure of LeNet

3 Results

3.1 CycleGan Performance

3.1.1 Image Quality

Examples of the images transformed by the CycleGAN from MNIST to SVHN are illustrated as figure 4. These examples demonstrate how effectively the CycleGAN has learned to adapt the distinct visual style of MNIST images to resemble those of the SVHN dataset. The images show significant transformation in terms of color and texture, which are key characteristics of SVHN.

Figure 4: Image transformed from MNIST to SVHN

3.1.2 Loss Metrics

The training process involved monitoring the loss functions for both the generators and discriminators. The following graphs display the generator and discriminator loss curves over the training epochs. Generator Loss explains how well the generator was able to fool the discriminator and is a measure of the quality of the generated images. Discriminator Loss measures the discriminator's ability to distinguish between real and fake images. A decrease in discriminator loss indicates improved discriminator performance.

3.1.3 Adam

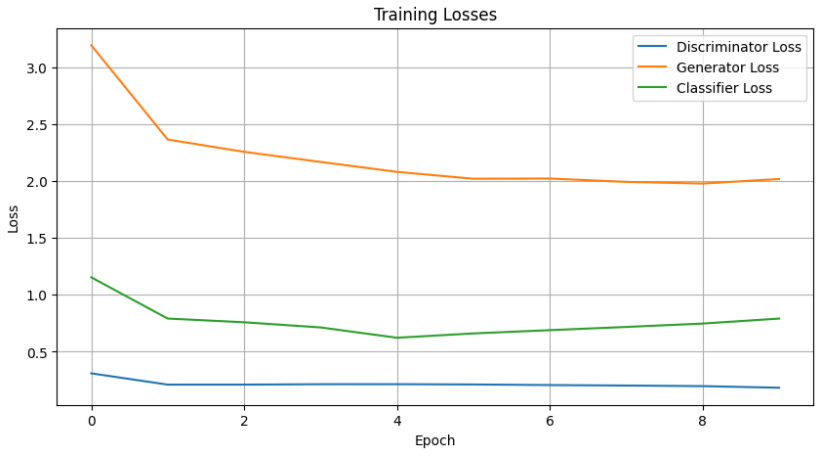

Figure 5: Loss functions with Adam optimizer

As we can see from the figure 5, Discriminator loss oscillates between 0.15 to 0.25 , which indicates that the performance of the discriminator is stable in distinguishing between real and generated images. The steady decrease shows that the training of the discriminator is effective and is able to distinguish between real and generated images better. And the stable decrease in Generator loss throughout the training process indicates the progress in the generator's ability to produce more realistic images. But Adam's converting speed is quicker than Adamax's.

3.1.4 Adamax

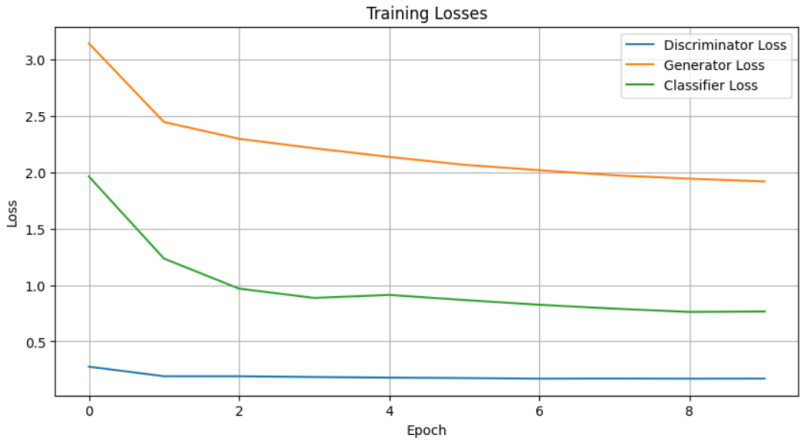

Figure 6: Loss functions with Adamax optimizer

As we can see from the figure 6, Discriminator loss oscillates between 0.14 to 0.2, which is more stable than the Adam one. It indicates that the model is able to distinguish between real and generated images more reliably during the training process, thus improving the efficiency and stability of training. The smoother descent curve of Generator loss represents a gradual improvement in the quality of the images produced by the generator during training and a relatively stable training process. This indicates that the generator is able to produce more realistic images more efficiently without large fluctuations or unstable training situations. However, there are still some fluctuations, suggesting the need for further adjustments to training parameters or model architecture to improve training stability and the quality of generated images.

3.2 Classifier Performance

3.2.1 Classifier Accuracy

- Adam

The classifier trained on transformed MNIST images with an Adam optimizer achieved an accuracy of 75.13% on the SVHN test dataset. This metric is crucial as it reflects the effectiveness of the domain adaptation performed by the CycleGAN and the generalizability of the classifier when applied to real-world SVHN images.

Figure 7: Prediction with Adam optimizer

The results indicate that while the CycleGAN is capable of transforming the style of MNIST images to closely mimic that of SVHN images, there are challenges in achieving high accuracy with the classifier. This could be attributed to the inherent differences in the datasets' distributions that may not be fully captured by the CycleGAN transformation. Furthermore, the classifier's architecture and training regime might require adjustments to better handle the features of the transformed images.

- Adamax

The classifier trained on transformed MNIST images with an Adamax optimizer achieved an accuracy of 80.11% on the SVHN test dataset. Adamax trains better than Adam probably because it uses the infinite norm of the gradient instead of the quadratic of the gradient, which may be more stable in

some cases and more robust to the choice of learning rate compared to Adam. The classifier still can't recognise similar looking numbers(such as 7 and 1) very well. It might be due to the dataset being still too simple to capture complex features adequately. Further increasing model capacity and employing data augmentation techniques could help address this issue.

Figure 8: Prediction with Adamax optimizer

3.2.2 Potential Improvements

- Enhancing CycleGAN Training: Adjusting the architecture or training parameters of the CycleGAN might result in higher quality image transformations.
- Advanced Classifier Architectures: Utilizing more complex neural network architectures that are more robust to variations in input data could improve classification accuracy.
- Data Augmentation Techniques: Implementing advanced data augmentation techniques, such as geometric transformations, color jittering, and mixup, can help improve the generalization and robustness of neural network models, especially when dealing with limited training data.

4 Conclusion

This project demonstrated the feasibility of using a CycleGAN for domain adaptation from MNIST to SVHN, effectively transforming the visual style of images to facilitate cross-domain classification. The CycleGAN was able to generate images that closely mimic the target domain, which were then used to train a LeNet classifier. Despite the complexities involved, the classifier achieved a respectable accuracy, showcasing the potential of GANs in unsupervised domain adaptation tasks.

Working on this project has been immensely educational, providing insights into the challenges and intricacies of deep learning models like GANs and the subtleties of domain adaptation. It reinforced the importance of fine-tuning model parameters and offered a hands-on experience with the potential and limitations of neural networks in artificial intelligence. The skills and understandings gained from this project are invaluable and will undoubtedly influence future projects, particularly in exploring advanced neural network architectures and their applications across different domains.

To enhance the model's performance, future work could explore:

- Integrating more robust discriminator architectures to improve the fidelity of the generated images.
- Employing different variations of GANs that might offer more stability and efficiency during training.
- Extending the model to multi-domain adaptation to handle more complex datasets.
- Experimenting with different loss functions that might capture the domain characteristics more effectively.

References

[1] 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, pp. 2223–2232.

[2] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[3] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, *et al.*, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, pp. 2672–2680, 2014.

[4] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.

[5] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7167–7176.

[6] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations*, 2014.

[7] C. Simon, *Mnist to svhn transfer with cyclegan*, https://github.com/cesimon/MNIST-to-SVHN, 2021.