CAR STYLE TRANSFER

by the image-to-image translation with CycleGAN

ABSTRACT

We apply CycleGAN to the two distinct cases of style transfer for images of cars. The first is color transfer (blue/red). The second is a normal/rusty pair transfer. We compare the results to different existing approaches of the proposed applications.

Keywords: style transfer, unpaired dataset, image-to-image translation, CycleGAN, color transformations.

Authors:
Frederico Santos – m20200604
Ivan Kisialiou – m20200998
Tiago Ramos – m20200613

NOVA IMS Deep Learning Course

Lisbon, April 2021

Introduction

Following the emergence of Generative Adversarial Networks (GAN) in 2014 [1], many approaches have been developed to succeed in a broad family of image-to-image (i2i) translation tasks. One of the state-of-the-art examples is CycleGAN, which expands GAN concept with a cycle-like architecture, which notably impinges a new set of constraints to the generators of the network. This approach was first introduced in 2017 by Zhu, J.Y. et al. [2], and its implementation can significantly broaden the range of i2i translation tasks that are currently achievable.

Moreover, the nature of the architecture incentivizes such a translation to be perfectly reversible, which means a model trained on the translation of an image of domain X to domain Y will also be proficient at the translation of an image of domain Y to X. The order and direction of the transfer is therefore irrelevant.

Task Definition and Approach

For this project, we applied the CycleGAN techniques, as described in [2], to two types of applications: blue to/from red cars (blue2red), and normal to/from rusty cars (normal2rusty).

For the first task, our motivation was to apply CycleGAN to a relatively simple problem, for which there can exist other solutions including some that are not reliant on neural networks at all (like color replacement with OpenCV) and compare the results.

For the second task, we wanted to explore the main proposed advantages of CycleGANs by choosing a type of style transfer that could satisfy the following criteria:

1) Not obvious but still recognizable translation, based on both color and texture changes
2) For which a paired image dataset would be reasonably difficult to produce
3) For which images of both domains would be reasonably easy to acquire

The first criterion stems directly from the capability of CycleGAN models to focus on middle-level features such as textures, colors and fill patterns, while retaining the major structural features of an image intact, even if a human may not objectively recognize what those are. The second comes from the suitability of the CycleGAN models to i2i translation tasks without the requirement to use sample pairs for training.

Background

The problem explored in the project is formally the i2i translation, i.e. generation of a new image in the domain Y from an original input image out of the domain X, combining both the features of the original image and the features of domain Y.

One of the first neural network methods that succeeded in the area of image-to-image translation was Neural Style Transfer algorithm [3], which utilizes the idea of constructing the target loss function in optimization task as a superposition of the content loss and style loss, which are calculated from CNN feature maps accounting for content representation of one image and style representation of the other. However, such an approach applies the style to the whole area of image, and not only to the object of interest.

Since the invention of GAN concept, it was applied to i2i translation problem in numerous works. The family of conditional GANs [4] yields impressive results. One can find the examples of application of this technique to such problems as Colormap/Street Scene, Colormap/Facade,

Satellite/Map, Day/Night, Sketch/Photo, Image Colorization, Resolution Increase, etc. However, this requires the usage of paired samples in the dataset. The need for paired images represents a heavy burden regarding the production of datasets for such tasks.

On the other hand, cyclic GAN algorithms like CycleGAN [2] or DiscoGAN [5] overcome this limitation. The leverage of two generator-discriminator pairs, one for each direction of transformation, together with cycle consistency loss replaces the role of pair matching in the dataset.

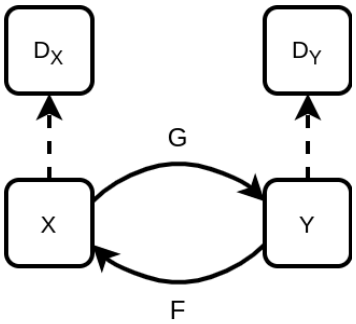
The idea evolves further, for example explorers propose the combinations of conditional and cyclic GANs with some portion of paired samples in unpaired dataset [6] or adding variational autoencoders to the architecture [7]. Some other solutions include the use of Transformers (attention networks) [8] but may appear too computationally heavy.

Related to the problem considered in our work, the color-to-color translation can be achieved with some traditional methods (without deep learning). However, getting high-quality and stable results may be challenging or even impossible, as such solutions lack generalization and often need additional tuning for every new input. At the same time, we have not found any examples of GAN approach to the problem of changing the color of cars.

Regarding the normal/rusty problem, the recent conference paper [9] shows how tricky can be a good solution.

Methodology

Approach to Architecture



To get the baseline, we implemented the OpenCV script which allows color conversion between blue and red cars. Our method included color thresholding/filtering in HSV space, background masking and finally the shift in Hue channel, corresponding to the difference between pure Red and Blue colors.

The architecture of the model is based on the published CycleGAN paper [2], which in turn adapts its architecture from Johnson et al. [10]. There are two generators, G and F, and two discriminators, D_X and D_Y. The architecture of both generators consists of an input layer, that takes 256x256 images, adds Reflective Padding to reduce artifacts, has a convolutional layer with kernel size 7 and stride 1 (naming convention followed Johnson et al., c7s1-64) that increases the image's features to 64. Two convolutional layers, with Instance Normalization [10] and ReLU activation, are added to decrease the image's width and height to 64 and increase the features to 256. Following the convolutional layers are 9 residual blocks [11] that keep the same shape of the tensor.

The idea behind using residual blocks is to transform the tensor but keep some of the original information, thus allowing more effective training of deep structures. This is achieved by taking X as input, going through two convolutional transformations into H(X), and summing X and H(X) at the end of the block.

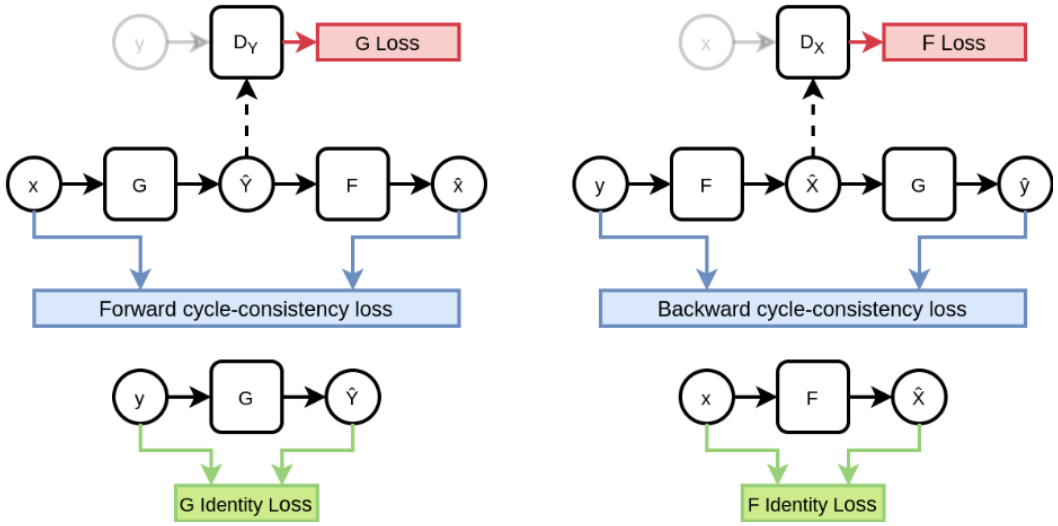
Two transposed convolutional layers increase the size of the image back to 256x256 pixels and 64 features, again with Instance Normalization instead of Batch Normalization and ReLU activations. Then, a final convolutional layer transforms the 64 features back into 3 with an activation function of tanh to recreate the image.

As for the discriminators, we apply the PatchGAN architecture [12], with LeakyReLU activations and Instance Normalization (except for the first layer). The network consists of 4 convolutional blocks, that outputs a 32x32 with 1 feature map image and this output is used to calculate the generators' loss.

We have included both network's architecture mapping into the source code package.

In our experiments, we tried several CNN and GAN architecture improvements. We experimented with different amounts of residual blocks, ranging from 3 to 12 (GPU constraints capped this amount at 9), implemented GlobalAveragePooling in the last layer of the discriminator to have the output be 1x1x1 instead of the PatchGAN architecture.

Approach to Loss



To keep color consistency between translations, identity mapping [13] was applied to the loss calculation. The loss of the generator is composed by three types of losses: Adversarial loss (G Loss + F Loss), Cycle-consistency loss (Forward cycle-consistency loss + Backward cycle-consistency loss) and identity loss (G Identity Loss + F Identity Loss).

Cycle-consistency loss can be understood as the difference between the original image x and the generated image x̂, after the image X translated to the domain Y and back to the domain X. Identity loss is the difference between the image X and the image X̂, translated by F, where it does not leave the domain X. This added identity loss allows the generator to keep color consistency when translating twice, from domain X to Y and back to X.

Considering that the CycleGAN paper was published in the year of 2017 (an eternity in the field of Deep Learning), we implemented the newer Rectified Adam (RAdam) [14] optimizer in all our networks and found substantial improvement over the Adam optimizer. RAdam improves the accuracy of the model and provides a faster convergence, which allowed us to experiment with

many different architectures. It is also important to note that the RAdam optimizer is much less sensitive to learning rate, so we mainly experimented with 10x increments and settled with the paper-suggested learning rate of 0.0002.

Datasets

For both tasks, we used the IMS Stanford Cars Dataset [15] to source images of both normal cars and colored cars, for which the labels were assigned manually upon human inspection. In total, about 620 images of blue cars were collected, and 1800 images of red cars were collected. A small test set of size 20 was split randomly for each class, so that each epoch trained with 600 images.

For the second task, we let the definition of ‘rusty’ retain its subjectivity, and sourced images for a dataset from Flickr with the search term ‘rusty car’, which returned a dataset of images classified by humans as a “rusty car”. About 500 images were collected, and after initial manual pruning for invalid images, 450 were used. The same number of pictures were sampled from the IMS Stanford Cars Dataset [15].

Preprocessing of images was done using the TensorFlow library capabilities. For the training set, preprocessing included resizing, random crop, jitter addition, and normalization. For the test set, only resizing and normalization were applied.

Evaluation Measures

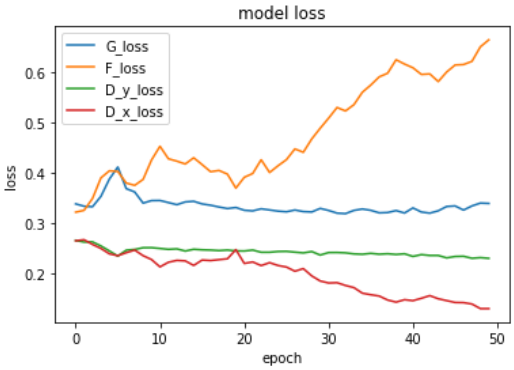
Besides the mathematical output of the model in terms of loss and training time, human evaluation will be naturally subjective. However, a few consensual guidelines can be set:

1) Is the model capable of identifying the parts of the car to be translated?
2) Is the model capable of generating a “blue/red” or “rusty-car” style in a picture?
3) Can the model transform a car in the picture in the described way?
4) Is the transformation of sufficiently good quality, i.e., clearly recognizable as one or the other style?
5) For the blue/red task, can the model outperform the output of the OpenCV baseline?

Results and Discussion

Blue2Red

To achieve these results, our CycleGAN model trained for 50 epochs, for a total of about 10 hours of train time. We present a chart of the loss per epoch below, as well as the resulting image output.



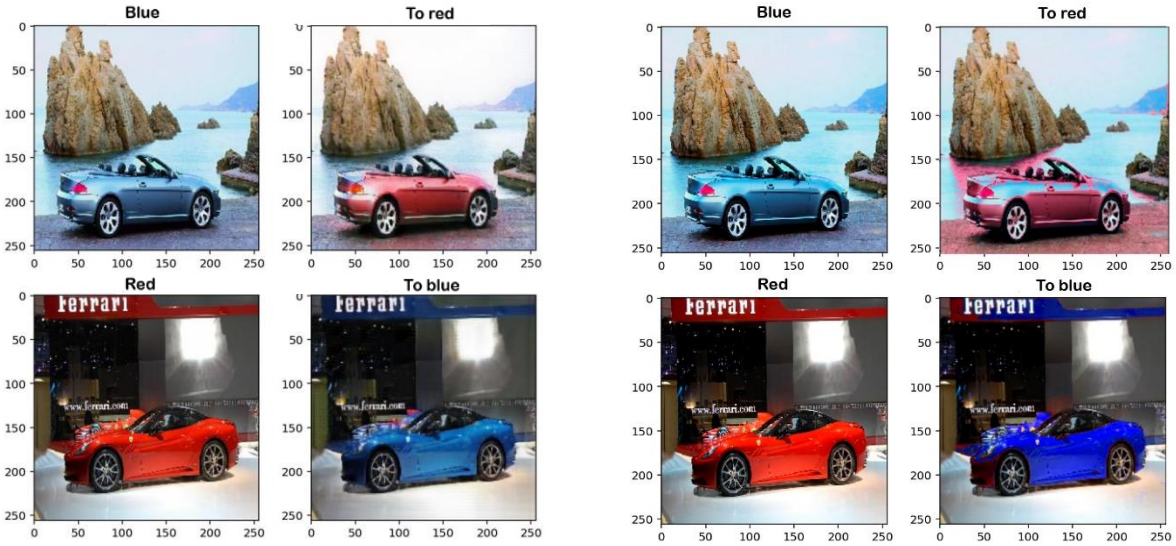
This plot is indicative of the overall improvement of the model over time, as well as of the imbalance of the dataset: F's loss comes from D_X, which seems to have overfit on knowing which images were real. This hypothesis is reinforced by the substantial decrease of D_X loss. Since the generators' losses are summed together before the gradient is calculated, this has less of an effect in the overall translation of the images, as seen below.

In terms of subjective appraisal, the outcome is positive. The model is capable of translating car color effectively, containing most of its transformation to the area of the car, without color spilling or overall loss of quality. In most cases, the model accurately identifies background elements as such, and keeps their texture and color, even if it is close in shade and tone to the car's. In some cases, for example when the color of the background element is too close to the car's and, this

element will be transformed as well. However, there are cases where the network successfully keeps the license plate's blue color and translates the entire car from blue to red.

In terms of the reconstruction of the original image, the model is overwhelmingly accurate at doing so, without incurring in significant shading changes or background information loss. It also succeeds in reconstructing background elements that had been changed from the first image, which is expected considering the built-in constraints of the architecture.

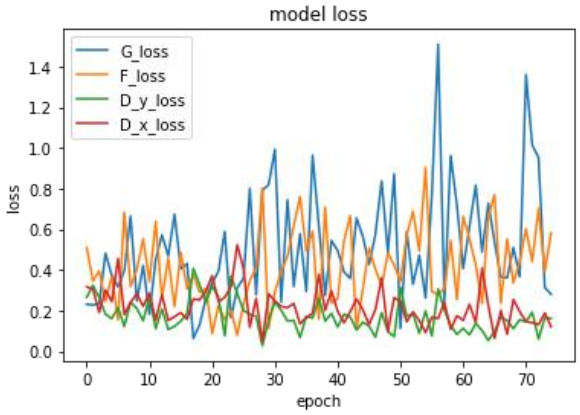
Comparison to OpenCV



The examples provided display the capabilities of the CycleGAN model (left) against the OpenCV output (right). CycleGAN correctly identifies and translates the color, matching its relative shade to a virtually indistinguishable transformation. It also succeeds at isolating the car in the picture, even when background of the same color is present. While it may still apply a color transformation to unrelated elements in the background, it does so effectively and completely, in contrast to OpenCV, which cannot do so without extensive per-photo tuning.

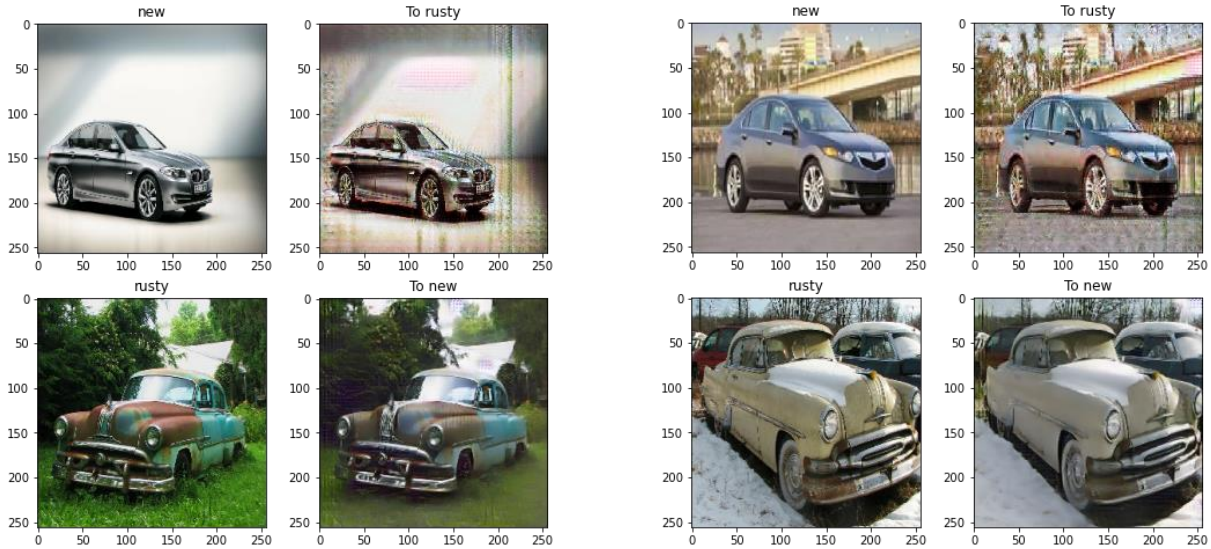
Normal2Rusty

The CycleGAN for the task trained with 9 residual blocks, 256*256 image size, and 75 epochs, with 450 images per epoch. We also present a transformation of such a model on some examples.



The first immediately noticeable aspect of the model's history is the spiking nature of the losses. While no loss truly stabilizes, as epochs increase, the learning process trades off generator losses in favor of reducing discriminator losses. The lower total average loss is achieved relatively early (about 4 to 5 epochs).

Regarding the outputted model, it appears that it does not distinguish between the car and its background – failing point 1 of our guidelines. Exploring further, it seems the model applies filters to the whole picture – in the normal to rusty case, the picture has a slightly darker texture, with colors one would typically associate with rust or degradation. In the opposite case, the picture is less saturated and sharp, which has the effect of removing artifacts from certain textures and making color along surfaces more consistent – a characteristic of newer and unused objects, such as cars.



These results imply the model is in fact capable of identifying something as subjective as "rusty", despite shallow specificity of its meaning, but it does not accurately identify, contour, and isolate the car itself, despite being able to do so in the color transfer task. The implications can be discussed from various perspectives:

1) Domain specificity. "Rusty" is a very broad and ambiguous search term which can return images that by themselves can represent many different specific styles [9]: everything from slightly aged to completely destroyed could arguably be classified as "rusty". Such a level of subjectivity increases the difficulty of a perfect reversible translation. The reversibility constraint of the CycleGAN becomes a detriment to this particular task.
2) Geometric Differences. Due to the architecture in the CycleGAN technique, the model is incentivized to retain the structure of the image, while being able to change its texture. Degraded cars, which fit the search term "rusty", can significantly differ in geometry to a car in a normal state, which the model does not transform. This type of limitation is signaled as such by the authors of the paper [2], and an important part of future work for CycleGANs.
3) Background and viewpoint. For the model to learn to apply textural changes to the whole picture implies that our intended translation target was not clear. This could be fixed with a better specified dataset. For example, sourcing the "normal" label to only new cars; specify the level of degradation to a narrower definition; constrain the datasets to a few viewpoints of the car.

4) Reverse translation ambiguity. The old-to-normal task can be mentally decomposed into the two independent tasks: (a) color translation of brown parts into the color of non-brown parts of original car, (b) texture translation old-to-normal. Task (a) fails because we have too few *partially* rusty cars in the dataset, and the model therefore is forced to convert the colors of whole cars. It almost ignores the task of taking the color of non-rusty parts and applying it to the whole car (again, because of the dataset). Moreover, in the color conversion part of the problem, cycle transform can be done via *any* intermediate color (brown-any-brown), simply because there are different colors in 'normal' dataset, and possible paths are averaged, which makes the network deal with textures only. Then the subtask (b) – textures – also fails, because the datasets are rather *'colorful' vs 'brown'* than *'polished' vs 'corrupted'*.

Therefore, we posit that the main challenge for this task is not the model itself, but rather the dataset it is supplied with – as is the case with many machine learning tasks.

Conclusion

We have confirmed the capabilities of CycleGANs by replicating its architecture and applying it to two image-to-image translation tasks.

For the blue2red task, we were able to achieve results comparable or overcoming the techniques that do not rely on GANs but have a different set of requirements which may constrain their application in situations where a CycleGAN is unencumbered, such as paired images datasets, or hardcoded boundary boxes and hue levels. This expands the set of possible solutions that can be used to tackle similar tasks.

For the normal2rusty task, we did not achieve favorable results. While we confirmed the limitations of the model in tasks where a style is characterized by different geometries, there is evidence our results were further obstructed by the dataset's quality as well as a shallow specification of the target domains.

References

[1] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y. (2014) Generative adversarial nets. In: Advances in neural information processing systems. 2672–2680

[2] Jun-Yan Zhu*, Taesung Park*, Phillip Isola, and Alexei A. Efros. (2017) Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks", in IEEE International Conference on Computer Vision (ICCV)

[3] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. (2016) Image style transfer using convolutional neural networks, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2414–2423.

[4] Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A. (2017) Image-to-image translation with conditional adversarial networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 5967–5976

[5] Kim, T., Cha, M., Kim, H., Lee, J., Kim, J. (2017) Learning to discover cross-domain relations with generative adversarial networks. arXiv preprint arXiv:1703.05192

[6] S. Tripathy, J. Kannala, and E. Rahtu. (2018) Learning image-to-image using paired and unpaired training samples. arXiv preprint arXiv:1805.03189

[7] Liu, M.Y., Breuel, T., Kautz, J. (2017) Unsupervised image-to-image translation networks. In: Advances in Neural Information Processing Systems, 700–708

[8] Ning Dai, Jianze Liang, Xipeng Qiu, and Xuanjing Huang. (2019) Style transformer: Unpaired text style transfer without disentangled latent representation. arXiv preprint arXiv:1905.05621

[9] Von Zuben, A., Nascimento, R. and Viana, F. (2020) “Visualizing Corrosion in Automobiles using Generative Adversarial Networks”, Annual Conference of the PHM Society, 12(1), p. 9

[10] D. Ulyanov, A. Vedaldi, and V. Lempitsky. (2016) Instance normalization: The missing ingredient for fast stylization. arXiv preprint arXiv:1607.08022.

[11] K. He, X. Zhang, S. Ren, and J. Sun. (2016) Deep residual learning for image recognition. In CVPR

[12] J. Johnson, A. Alahi, and L. Fei-Fei. (2016) Perceptual losses for real-time style transfer and super-resolution. In ECCV

[13] Y. Taigman, A. Polyak, and L. Wolf. (2017) Unsupervised cross-domain image generation. In ICLR

[14] Liyuan Liu , Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han (2020). On the Variance of the Adaptive Learning Rate and Beyond. the Eighth International Conference on Learning Representations. arXiv:1908.03265

[15] J. Krause, M. Stark, Jia Deng, Li Fei-Fei. (2013) 3D Object Representations for Fine-Grained Categorization, IEEE Workshop on 3D Representation and Recognition, at ICCV 2013 (3dRR-13). Sydney, Australia.

Appendix

The following are additional examples of the output of the blue2red model.