Project CV2: Image-to-image translation with Pix2Pix conditional GAN architecture

Stefano Iannicelli[1,†] **and Ettore Caputo**[2,†]

[1]*MSc student at Unical, IT - Erasmus student at UDC, ES*
[2]*MSc student at Unical, IT - Erasmus student at UDC, ES*
[†]*These authors contributed equally to this work*

This manuscript was compile on October 10, 2024

Abstract

We investigate how the Pix2Pix model works on an image-to-image translation task, translating a label map to obtain a realistic image. This work show that Pix2Pix model is able to learn to synthesize photos from label maps, reconstruct objects from edge maps, and colorize images. The process is been focused on the TU-Graz dataset that contains 400 photo taken by a drone. By doing some improvements, changing loss, using augmentation and a combination of both permit to obtain better results (by comparing performance metrics).

Keywords: Pix2Pix, Image-to-Image, TU-Graz

INTRODUCTION

Transform a label map in realistic images can be usefull in some case, like translating the English in Spanish. The Pix2Pix model proposed by Isola et al. [3] proved that this approach is effective at synthesizing photos from label maps. We started from a pre-existing implementation of Pix2Pix model from GitHub[1]. This is the same network implemented in [3] but we chosen this because was simpler to understand and represent a minimal version compered with the original repository[2]. We trained this model on a ten times down-scaled version of the TU-Graz[3] dataset.

PIX2PIX MODEL

The problem is to generate realistic images starting from input label maps, the dataset used is the TU-gratz which contains 400 aerial photos taken with drones at a resolution of 600×400. To do this, the pix2pix model was used which is made up of a conditional Generative Adversarial Network (cGan). A GAN is used, since modeling a loss function suitable for the problem of image generation is very difficult, therefore a network called a discriminator is used which will learn the loss function and a generator network to generate the images.
The architecture of the generating network is an encoder-decoder with the addition of skip connections, this is called U-net, the skip connections are connections between layers, each layer i is connected to layer n-i this is done because in the classic architectures encoder decoder creates a bottleneck that causes difficulty in generating images. The discriminator network is based on the assumption of the Markovian process of the images, in fact it is assumed that the pixels of the images are independent between one patch and another, therefore the discriminator network is very small because it only takes small portions as input (Patch size N×N), each patch is fed to a convolutional network which calculates the loss and finally the average is taken across all. This allows numerous advantages, one of which is the reduction in the number of weights in the network. The objective of a conditional GAN can be expressed as:

$$L_{cGAN}(G,D) = E_{x,y}[logD(x,y)] + E_{x,z}[log(1-D(x,G(x,z))] \quad (1)$$

where G tries to minimize this objective against an adversarial D that tries to maximize it. To improve the images, the l1 norm was added to the loss, which allows the blurring to be reduced. The final objective is:

$$G^* = arg \max_G \min_D L_{cGAN}(G,D) + \lambda L_{L1}(G) \quad (2)$$

1. ARCHITECTURE

The generating network takes the label maps as input and returns the reconstructed image, to do this it uses an encoder and a decoder, respectively made up of 8 convolutions and 8 inverse convolutions, each layer has batch normalization and uses the Leaky ReLU as the activation function. Finally, a tensor obtained from the last layer is returned in output by applying the tanh function to it. The discriminator network is made up of 5 convolutions and in output it returns a scalar which represents the loss value for the input image (see Figure 1).

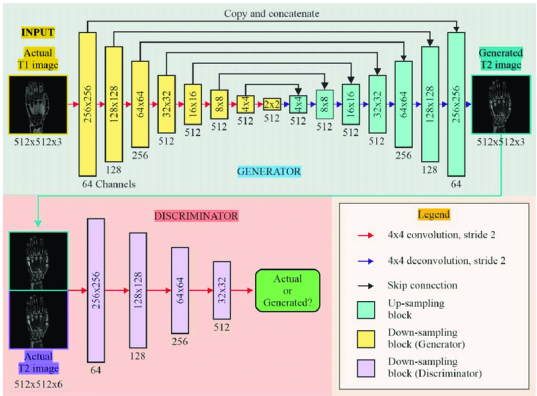

Figure 1. Pix2Pix network architecture

2. BASELINE APPROACH

To obtain the first results we modified and added some code to the github repository that we used as a starting point. First we created the dataset, in fact it is made up of 1200×400 images, in each of them there are two 600×400 images which represent the input image and the ground truth, so we divided the original image through Python's PIL library. Furthermore, we have created two test scripts, in the first we calculate all the metrics on the 50 test images, while in the second the images produced by the network through matplotlib are shown (see Figure 2).

[1]https://github.com/akanametov/pix2pix
[2]https://github.com/phillipi/pix2pix
[3]https://www.tugraz.at/institute/icg/research/team-fraundorfer/software-media/dronedataset

3. IMPROVEMENTS

In order to improve what has been done in the baseline approach we started by increasing the dataset dimension by performing online data augmentation. After that we tried with a different loss function and then a combination of both techniques.

3.1. Data augmentation

The images has been transformed by rotation, cropping, horizontal and vertical flip.

3.2. Different loss functions

We tried a different loss function [4] in order to get better results. For this reason we tried to add the *Kullback–Leibler* divergence in a sum with the original Pix2Pix loss.

In mathematical statistics, the *Kullback–Leibler* (KL) divergence, denoted $KL(P \,||\, Q)$, is a type of statistical distance that measure of how one probability distribution P is different from a second, reference probability distribution Q.

To compute the *Kullback–Leibler* divergence we used the function already implemented in torch defined as follow:

$$KL(y_{pred}, y_{true}) = y_{true} \cdot \log \frac{y_{true}}{y_{pred}} = y_{true} \cdot (\log y_{true} - \log y_{pred}) \quad (3)$$

The final loss is given by the sum of the previous loss plus the divergence of KL multiplied by a beta factor $\beta \in [0, 1]$. We chose beta equal to 0.2 as the KLD has very high values, by doing so we reduced its contribution, preventing the minimization algorithm from focusing only on the KLD and not on the rest of the loss.

$$L_{cGAN} = L_{cGAN} + \lambda \cdot L_{L1} + \beta \cdot KL \quad (4)$$

4. TRAINING

The training phase is equal for each configuration described before, then it is the same for the base model, the model with augmentation, the base model with KLD loss and so on. Therefore to the train phase we divide the dataset in two parts, the train set and the test set, respectively 87,5%(=350 images) and 12,5%(=50 images). The training process is based on 500 epochs, no changed has been made to the hyper-parameters values from the original paper[4]. The images has been down-scaled from 600x400 to 256x256 according to the input size in the original version of Pix2Pix model, furthermore this increased the train speed, in fact to perform the 500 training epochs the model took approximately 2 hours on an Nvidia P100 gpu provided by kaggle free cloud.

4.1. The algorithm

Algorithm 1 Train

```
for epoch=1 , ... , N_eph do
    for real, l_map ∈ TrainSet do
        fake ← generate(l_map)
        fake_pred ← discriminate(fake, l_map)

        fake_pred ← descriminate(fake, l_map)
        real_pred ← descriminate(real, l_map)

        update_generator()
        update_discriminator()
    end for
end for
```

[4] lr = 0.002, $\beta_1 = 0.5$, $\beta_2 = 0.999$

5. METRICS

5.1. VIF - Visual Information Fidelity

VIF is a metric usually bounded between [0, 1]. When the test image is an exact copy of the reference image, VIF is exactly unity. Furthermore, and this is where VIF has a distinction over traditional QA methods, a linear contrast enhancement of the reference image that does not add noise would result in a VIF value larger than unity, signifying that the contrast-enhanced image has a visual quality superior to the reference image. Then a higher value means better results.

5.2. UQI - Universal Quality Index

The Universal Quality Image Index (UQI)[1] was created by modeling an image distortion as a combination of loss of correlation, distortion of luminance, and contrast. The UQI has a range from 0 to 1. An image with a UQI of 1 has a high quality. An image with a lot of distortion will have a low UQI.

5.3. SSIM - Structural Similarity Index Measure

The structural similarity index measure (SSIM)[2] is the successor of the UQI metric. SSIM is an image quality metric, it is computed for the image with respect to the reference image. The reference image is usually needs to be of perfect quality. This quantitative measure considers three parameters namely luminance, contrast and structural information between the two images to computed the SSIM value. The difference with other techniques such as MSE or PSNR is that these approaches estimate absolute errors. Structural information is the idea that the pixels have strong inter-dependencies especially when they are spatially close.

5.4. PSNR - Peak Signal Noise Ratio

Peak signal-to-noise ratio (PSNR) is the ratio between the maximum possible power of an image and the power of corrupting noise that affects the quality of its representation. To estimate the PSNR of an image, it is necessary to compare that image to an ideal clean image with the maximum possible power. The PSNR value is expressed in decibels (dB) and provides a numerical value that quantifies the similarity between the two images. When the PSNR value is high, it indicates a higher level of similarity and better image quality between the original and processed images. A high PSNR value implies that the processed image has minimal distortion or noise compared to the original image.

RESULTS

According to what we said before, to evaluate the models we used some metrics. In the Table 1 are reported the results for each configuration. The augmented model with Kullback–Leibler divergence achieve better results than the others in all metrics. The metrics were computed on the entire test set, for this reason we can say that the model (aug + kld) return more realistic results with more deep colors and details than the others we tried. What has been said is demonstrated by the Figure 2 where we reported six examples about the output generation from each model. In fact the last column, on the right-side, compared with the others it is generally sharper and more precise, which makes the image more realistic.

Table 1. Performance metrics result

Model	**Vif**	**Uqi**	**SSim**	**PSNR** (dB)
base	0.126	0.039	0.187	13.400
base + kld	0.113	0.045	0.226	13.643
aug	0.153	0.044	0.236	13.808
aug + kld	**0.183**	**0.053**	**0.242**	**13.894**

Figure 2. Some outputs results from each trained model. Each row represent a test example, starting from left, the first image is the input image, this will be processed by the generator from each model and the output from everyone, will be plotted from the third column. The second image in each row represent the ground truth used to compute each metric by compering it with the generator output.

CONCLUSION

In this project we have used the Pix2Pix model to resolve a image-to-image task on the TU-Graz dataset. At the end of the training we evaluated various models by computing different metrics. The best model trained in this work, is the result of the combination of two approaches: by augmenting and by adding a new term in the loss function, in our case the Kullback–Leibler divergence. Thus, the results in this work suggest that using this two approaches can promise better performance.

REFERENCES

[1] Z. Wang and A. Bovik, *A universal image quality index*, 2002. DOI: 10.1109/97.995823.

[2] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, *Image quality assessment: From error visibility to structural similarity*, 2004. DOI: 10.1109/TIP.2003.819861.

[3] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, *Image-to-image translation with conditional adversarial networks*, 2018. arXiv: 1611.07004 [cs.CV].

[4] A. Abu-Srhan, M. A. Abushariah, and O. S. Al-Kadi, *The effect of loss function on conditional generative adversarial networks*, 2022. DOI: https://doi.org/10.1016/j.jksuci.2022.02.018.