Journal of Computer Science
https://dergipark.org.tr/en/pub/bbd

ISSN, e-ISSN: 2548-1304
Volume: 7, Issue: 1, pp: 1-9, 2022
https://doi.org/10.53070/bbd.1019451
Research Paper

Automatic Mustache Pattern Production on Denim Fabric with Generative Adversarial Networks

Emrullah Şahin*[1], Muhammed Fatih Talu[2]

[1]Computer Engineering Department, Dumlupınar University, Kütahya / Türkiye

[2]Computer Engineering Department, İnönü University, Malatya / Türkiye

(essahin95@gmail.com, fatih.talu@inonu.edu.tr)

Received: Nov.5, 2021 Accepted: Dec. 3, 2021 Published: Jun. 6, 2022

Abstract— Mustache patterns drawn on denim jeans are created with a laser beam device. For this device to draw the desired mustache pattern, the pattern visual must be prepared. For the mustache patterns in the sample jeans taken from the customer to be transferred to the visual, specialist personnel in the Photoshop program should perform an average of 2-3 hours of work. This situation causes the production speed to slow down and human errors occur. In this study, a new approach is proposed that automatically produces the pattern visual by detecting the mustache patterns in the sample jeans samples to be taken from the customer. In this approach, an updated version of the Pix2Pix architecture within the Generative adversarial network (GAN) is used to produce mustache pattern images. The training was carried out with a dataset constructed from jeans and mustache pattern images, and the production of different mustache patterns depending on the personnel was prevented. As a result of the experimental studies, while the production speed of the mustache pattern fell below one second, it is seen that the production accuracy is around 89%. The next study, it is aimed to increase the accuracy by ensuring the standardization of the images in the dataset.

Keywords: Mustache pattern, Laser engraving, Denim Fabric, Generative adversarial networks

1. Introduction

The fact that denim fabrics can be used easily in all areas of life causes the demand to increase day by day and pushes the fabric manufacturers to take new actions to meet the increasing demand. Depending on the development of technology, issues such as decreasing personnel expenses and production costs in denim fabric production, and rapid production of products with high comfort and quality become increasingly important for businesses [1-3].

One of the reflections of the developments in the fashion world to the ready-to-wear industry is the mustache patterns drawn on denim fabrics. Mustache patterns add a different style to the plain appearance of the fabric.

Laser engraving machines are shown in Figure 1. (a) is used to draw mustache patterns on the produced plain denim fabrics. These machines can burn laser light ($\lambda = 10640$nm) on the surface of the denim fabric, which is placed in an area of approximately 2m^2, with very precise sizes such as 0.01mm. Configuration parameters in the combustion process are realized with the software shown in Figure 1. (b) This software takes the image of the mustache pattern to be drawn on the fabric as input and transfers it to the fabric in the specified configuration. For the mustache pattern images to be input to the laser engraving machine, either a new design is made or the mustache effect is extracted from the image taken from a ready-made sample (mustache denim fabric). For this reason, there is a need for personnel specialized in visual editing programs such as Photoshop. It takes about 2-3 hours of work for experts to construct a mustache pattern visual. Making new mustache designs is a relatively easy process to imitate the existing mustache model. Because the desired effects can be easily created on a plain denim fabric image. However, the process of working on the real images obtained after the samples supplied from the customer are photographed and the process of removing the mustache effects is very complicated as it requires the application of many image processing operations. In addition, the output (mustache pattern image) obtained according to the experience of the expert varies. It is a very sensitive process since a small error or an undesirable effect in the produced mustache pattern visual will be reflected on the product.

(a) **(b)**

Figure 1. Laser engraving system: **(a)** Machine, **(b)** Software [4]

For these reasons, there is a need for a system that can automatically and objectively produce mustache patterns from denim fabric. As a result of the researches, no study in the literature can automatically produce the mustache pattern. In this study, a prototype has been prepared that can automatically produce mustache patterns with generative adversarial networks.

Generative adversarial networks emerged in 2014 by Goodfellow and his team [5]. This architecture, which was developed based on the min-max algorithm, is a system in which learning occurs by the opposite operation of two convolutional models [5].

The DCGAN architecture, which converts a simple gaussian signal to an image by deconvolution to produce synthetic images similar to samples in a dataset, was developed by Radford et al. in 2015 [6]. To synthesize low-resolution images at high resolution, the perceptual similarity cost-based SRGAN architecture was developed by Ledig et al. in 2016 [7].

CycleGAN architecture based on a two-sided authentication process was developed by Zhu et al. in 2017 to perform image-to-image translation on unsupervised datasets [8].

Progressive GAN architecture was developed by Karras et al. in 2017 to produce high-resolution quality artificial images. This architecture can synthesize 1024x1024 images by learning to produce realistic images over time after starting the training with 4x4 images [9].

The spatially adaptive normalization-based SPADE architecture for synthesizing semantic map-based images was developed by Park et al. in 2019 [10].

The Real-ESRGAN architecture was developed by Wang et al in 2021 to be able to synthesize low-quality images with high quality. The main feature of this architecture is that the network can focus on more than one target at the same time even if an image contains multiple problem areas such as low resolution, blur, compression, or noise [11].

2. Material and Method

2.1 Generative Adversarial Networks

Introduced in 2014 and whose architecture is shown in Figure 2. generative adversarial networks are a model that is based on the game theory of the famous mathematician John Nash and consists of two network structures that learn by working against each other. One of the networks is called a generator and the other is called a discriminator. While the generative network is trying to produce more and more real-like artificial (fake) images, the discriminator network is a simple binary classifier that seeks to distinguish between real and fake images. While the producer creates fake images to fool the discriminator, the discriminator tries not to be deceived between the fake and the real image by learning over time [5,12,13].

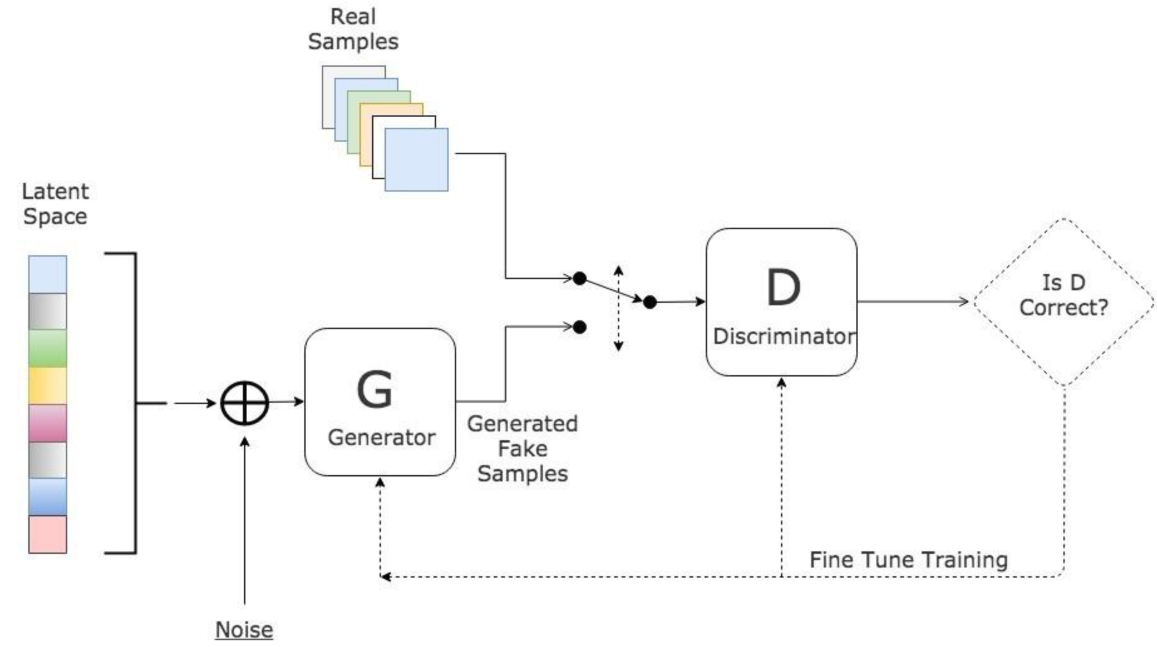

Figure 2. General schema of GAN architecture [14]

Weights and bias values of generator and discriminant networks are calculated with the backpropagation algorithm. Figure 3. shows in stages how the GAN architecture learned to generate a simple Gaussian signal. The generating network receives a randomly generated z signal as input and ensures that the green-colored signal is generated. The discriminator tries to distinguish between real (black) and artificial (green) signals. The blue signal indicates the error loss of the splitter.

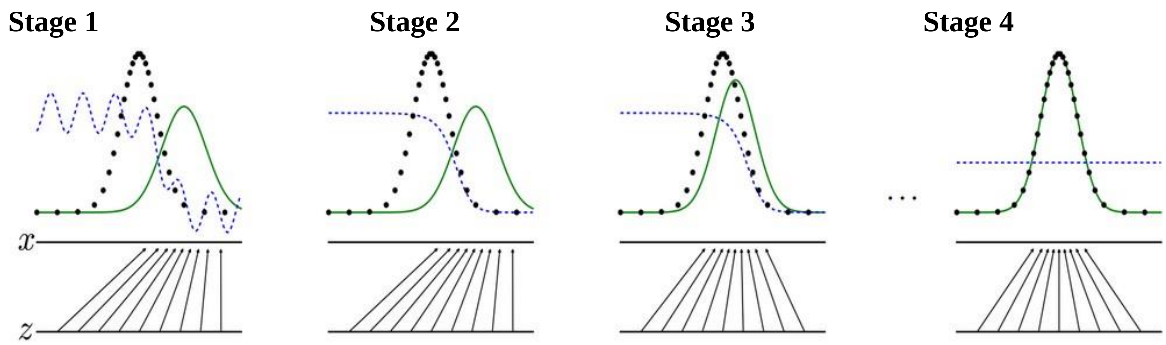

Figure 3. Gradual learning of GAN: Black true, Green generated, Blue discriminator error signals [5]

When the GAN architectures in the literature are examined, it is seen that there are different types of signal from the signal, image from the signal, and image from the image [13-15]. This study, since it is aimed to obtain a mustache pattern image from the sample denim fabric visual, it is focused on architectures that can produce images from the image. Pix2Pix [16,17] is seen to be one of the most frequently used in this type of architecture.

2.2. Pix2Pix

Pix2Pix is an architecture that can translate from image to image. This architecture was presented in 2016 by researchers from Berkeley in their work “Image-to-Image Translation with Conditional Adversarial Networks.” Most of the problems in image processing and computer vision can be posed as “translating” an input image into a corresponding output image [17]. For example, a scene may be rendered as an RGB image, an edge map, a semantic label map, etc. Thanks to the Pix2Pix architecture, an important step was taken in image-to-image translation on matched image pairs.

The cost value in classical conditional GAN architectures is expressed in Equation (1):

$$\mathcal{L}_{cGAN}(G, D) = \mathbb{E}_{x,y}[\log D(x, y)] + \mathbb{E}_{x,z}\big[\log D\big(x, G(x, z)\big)\big] \tag{1}$$

Accordingly, while G network wants to minimize this objective function, D wants to maximize the network. Thus, the cost of the optimal producer is written as in Equation (2):

$$\arg min_G \, max_D \, \mathcal{L}_{cGAN}(G, D) \tag{2}$$

Pix2Pix conditional GAN architectures is a version and adds the cost of similarity between the real image and the fake image to the cost function expressed in Equation (1). This cost value is calculated using the L1 distance metric, shown in Equation (3).

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y,z}[||\,\mathcal{Y} - G(x, z)\,||_1] \tag{3}$$

As a result of adding L1 cost, Pix2Pix the cost function is defined as Equation (4).

$$G^* = \arg min_G \, max_D \, \mathcal{L}_{cGAN}(G, D) + \lambda \, \mathcal{L}_{L1}(G) \tag{4}$$

The general architecture of Pix2Pix is shown in Figure 4. The architecture includes 2-dimensional convolution (Conv2d), Leaky Relu, and Batch Normalization layers [17-19]. Since this study aims to extract mustache patterns from denim images, denim images were given to the architectural entrance and mustache images to the output. The 256x256x3 size denim fabric at the entrance is encoded and reduced to the 1x1x512 feature vector in the middle, and again the feature vector is decoded and mapped to the same size mustache pattern image.

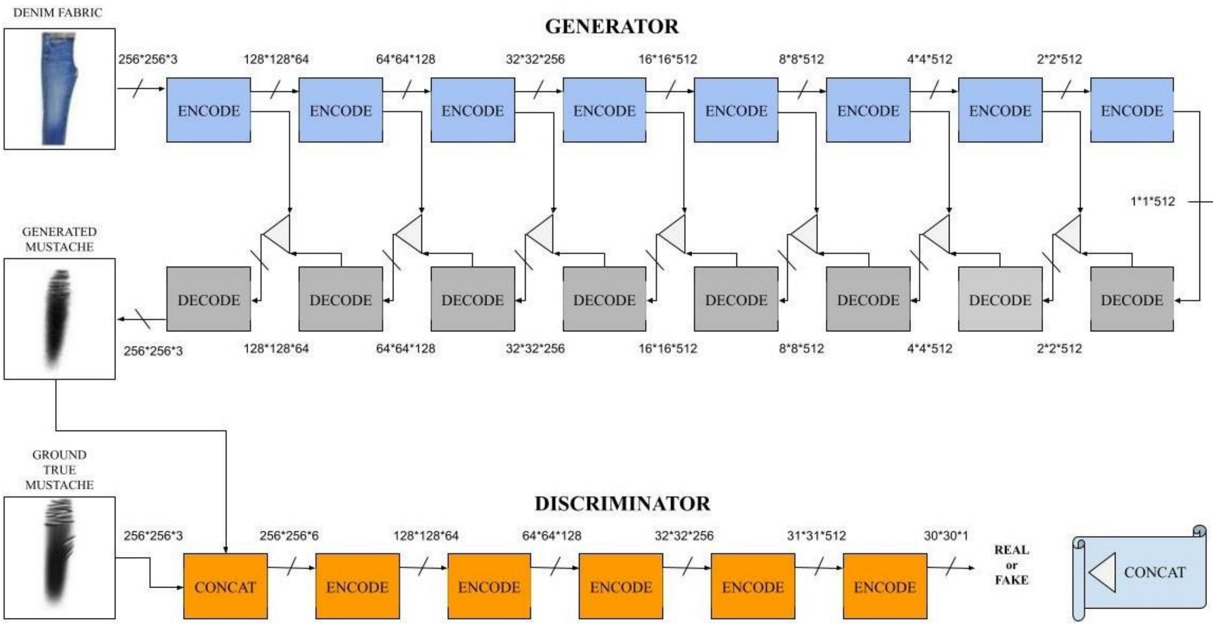

Figure 4. The use of Pix2Pix architecture in mustache pattern extraction

2.3 Proposed Method

Generative adversarial networks can memorize the information of the images in the training dataset while translating from one image area to another during the training phase. This problem is referred to as over-fitting in the literature. Due to excessive learning during the training phase, the desired quality image production cannot be achieved in the testing phase. To avoid this problem, regularization is applied to some layers of the network or the cost function. To prevent this problem, changes were made to the main structure of the generator network in the Pix2Pix architecture. By default, the encoder of the generator network has Convolution (Conv), Batch Normalization (Batch Norm), and Leaky Relu layers, while the decoder has Deconvolution (Deconv), Batch Normalization, and Relu layers. In the updated version, the Dropout layer has been added to the last 4 blocks of the encoder and the first 4 blocks of the decoder, making the network more durable to over-fitting. Thus, it was ensured that it produced quality and noise-free images during the test phase. The proposed method is named Pix2Pix v2. The schematic of this architecture is shown in Figure 5.

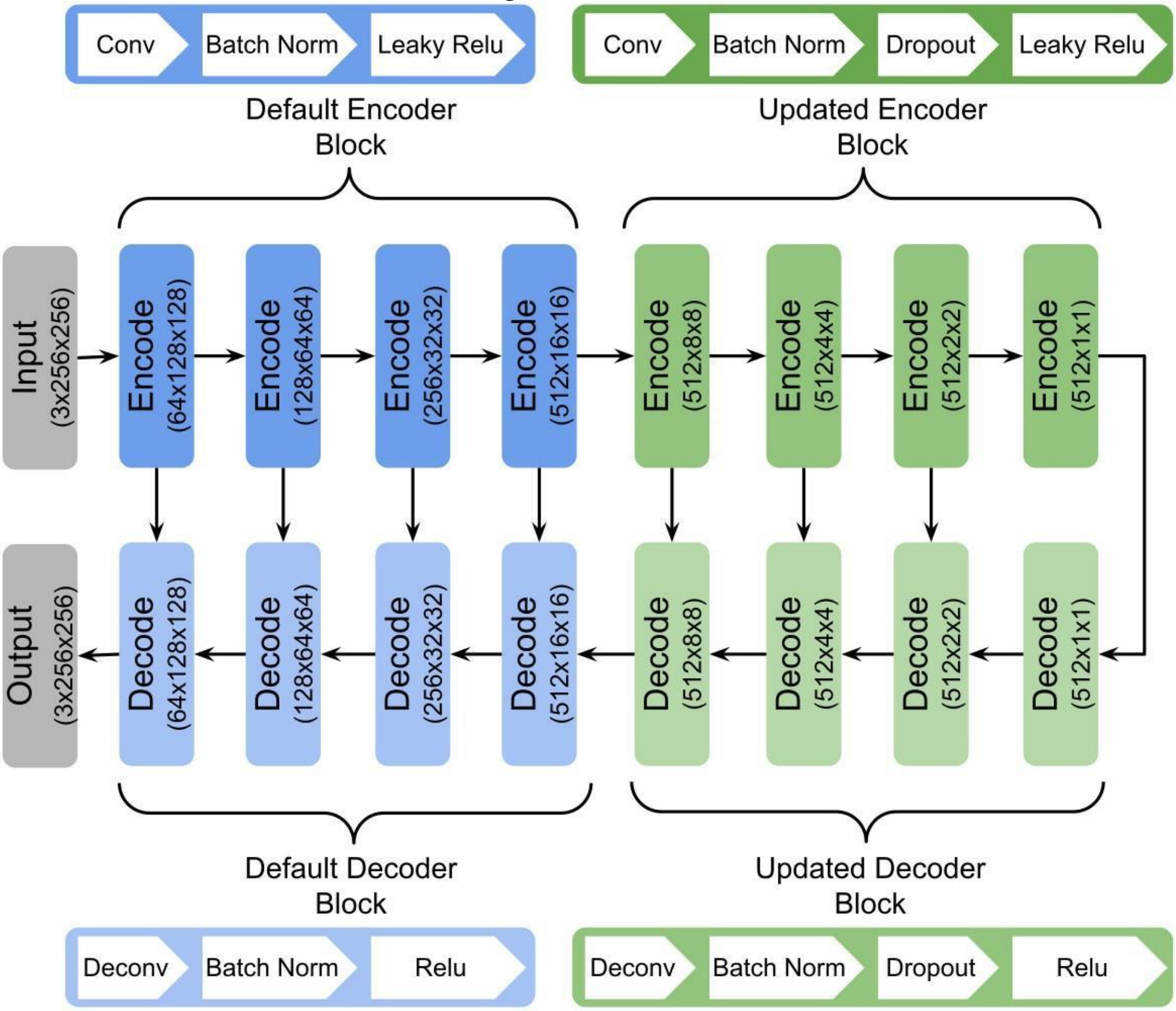

Figure 5. Updated version of Pix2Pix generator network architecture

2.4 Denim2Mustache Images Dataset

This work was carried out in Baykan Denim A.Ş.'s production factory in Malatya. In the construction of the dataset, the denim fabric sample images obtained by the company so far and the manually drawn mustache images based on these were used. A total of 976 denim-mustache image pairs were obtained. The dataset includes images of three different denim fabric types, including pants, skirts, and shorts, from different regions such as front, back, and hem.

It has been observed that these images of the employees of the company, which were obtained with their mobile phones, do not have a standard form. To put it more clearly, it has been observed that the background views, light distributions, sizes, and perspectives of the fabric images are different. To transform the data into a standard form, a manual cleaning study was carried out on each denim fabric image. In this study, the foreground object (fabric image) was extracted from the background and placed on a white background in the center of the image. A similar process was done in whisker patterns so that the input and output dimensions are similar (256x256x3). In addition, by normalizing the visuals, it was ensured that the differences arising from the light distribution were eliminated. A few denim fabrics and their mustache pattern images in the dataset as a result of pre-processing are shown in Figure 6.

Figure 6. Image samples of denim fabric and mustache patterns in the dataset were obtained as a result of pre-processes.

2.5 Similarity Metrics

In this section, similarity metrics used in the study are explained.

2.5.1 SSIM

The structural similarity metric (SSIM) is a metric that analyzes the perceived change in the image along with important perceptual properties such as brightness, masking, and contrast. This metric is a statistical measurement built on the mean ($\boldsymbol{\mu}$) and standard deviation ($\boldsymbol{\sigma}$) parameters in calculating the similarity between the image pairs, ignoring the positional difference between the pixels in the image [20].

Calculation of structural similarity between the statistically real ($\boldsymbol{x}$) and fake ($\boldsymbol{y}$) image is shown in Equation 5. In this equation, $\boldsymbol{\mu_x}$ and $\boldsymbol{\mu_y}$ denote the pixel mean of the actual and fake image, the variance of $\boldsymbol{\sigma_x^2}$ and $\boldsymbol{\sigma_y^2}$, while $\boldsymbol{\sigma_{xy}}$ denotes the covariance between the actual and fake image. In addition, the constant values $\boldsymbol{c_1} = (\boldsymbol{k_1 L})^{\boldsymbol{2}}$ and $\boldsymbol{c_2} = (\boldsymbol{k_2 L})^{\boldsymbol{2}}$ are calculated by taking the $\boldsymbol{L}$ value 255., which specifies the pixel pitch.

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{({\mu_x}^2 + {\mu_y}^2 + c_1)({\sigma_x}^2 + {\sigma_y}^2 + c_2)} \qquad (5)$$

2.5.2 Accuracy Score

To calculate the similarity cost between the two image samples, the equality of the pixel value in the real ($\boldsymbol{x}$) and fake ($\boldsymbol{y}$) image is checked [21]. If the values are equal, they are summed, returning 1 otherwise 0. Divide the total result by the number of pixels ($\boldsymbol{n}$). The main formula of the metric is shown in Equation 6.

$$Acc(x,y) = \frac{1}{n}\sum_{i=1}^{n} 1(x_i = y_i) \qquad (6)$$

2.5.3. MAE

The similarity cost between two image samples is calculated by taking the absolute value of the point error [22]. The main formula for this metric is shown in Equation 7. In the equation, the absolute value of each pixel difference between the real ($\boldsymbol{x}$) and fake ($\boldsymbol{y}$) image is summed. Divide the total result by the number of pixels ($\boldsymbol{n}$). While measuring, the direction of the error is not taken into account, it focuses on the absolute difference.

$$MAE(x,y) = \frac{1}{n}\sum_{i=1}^{n} |x_i - y_i| \qquad (7)$$

3. Results

The training process of the data set of denim2mustache visuals was carried out on a computer with a GTX 1660 Titan graphics card. The training process, which took place at 300 epochs, took approximately 15 hours. Python programming language and Keras library were used for training and testing. The data set is divided into 90% training and 10% testing. After the first training of Pix2Pix architecture was completed, the missing functions were examined and it was seen that the test set had a high loss, although the loss of the training set was very low. These results show that the original version of the network is over-fitting. Dropout operation was carried out in the last 4 layers of the encoder, and in the first 4 layers of the decoder, network to prevent excessive learning. In Table 1. the results of the accuracy rates of the original and updated Pix2Pix architectures are given. The original Pix2Pix architecture results are displayed as v1, while the updated architecture is expressed as v2.

Table 1. Training and test result of the original and updated Pix2Pix architecture

Model Name	**Metric Name**		
	SSIM ↑	**Accuracy Score ↑**	**MAE ↓**
Pix2Pix v1 (Testing)	0.86	0.58	72.02
Pix2Pix v1 (Training)	0.94	0.66	55.54
Pix2Pix v2 (Testing)	**0.89**	**0.63**	**61.53**
Pix2Pix v2 (Training)	**0.92**	**0.70**	**40.96**

After the training was completed, test results were obtained for both architects. Test results of several denim fabric visuals are shown in Figure 7. Accordingly, it has been observed that the visuals obtained with Pix2Pix v1 architecture contain high levels of noise. It has been observed that significant noise is eliminated with the updated Pix2pix v2 architecture. However, it is seen that the targeted real mustache pattern image cannot be produced in the desired similarity yet. The biggest reason for this is that the images in the dataset are not shot to a certain standard. The next study will focus on this standardization process.

4. Discussion and Conclusion

In this article, a study has been conducted on the automatic generation of mustache patterns on denim fabric images. Thus, mustache images are produced quickly and accurately without the need for manual drawings (approximately 2-3 hours) made by expert personnel looking at the denim fabrics, and in the next stage, it is ensured that plain denim fabrics are made with a mustache by giving them to the laser beam device. In this study, a dataset containing denim fabric images and mustache pattern models is constructed for automatic production. This dataset was used in training a generative adversarial network architecture known as Pix2Pix. As a result of the experimental studies, it has been observed that classical architecture provides excessive learning and produces unwanted noisy results on test data. As a result of the update on the architecture, it has been observed that excessive learning has been eliminated and the noise effect has been cleared. In this study, in which very promising results were obtained, work on the standardization of the dataset continues.

Acknowledgment

This study was funded by the “Bilimsel Araştırma ve Koordinasyon Birimi” of İnönü University with the project number “FKP-2021-2144”. We would like to thank Baykan Denim A. Ş and İnönü University for making the dataset used in the study accessible.

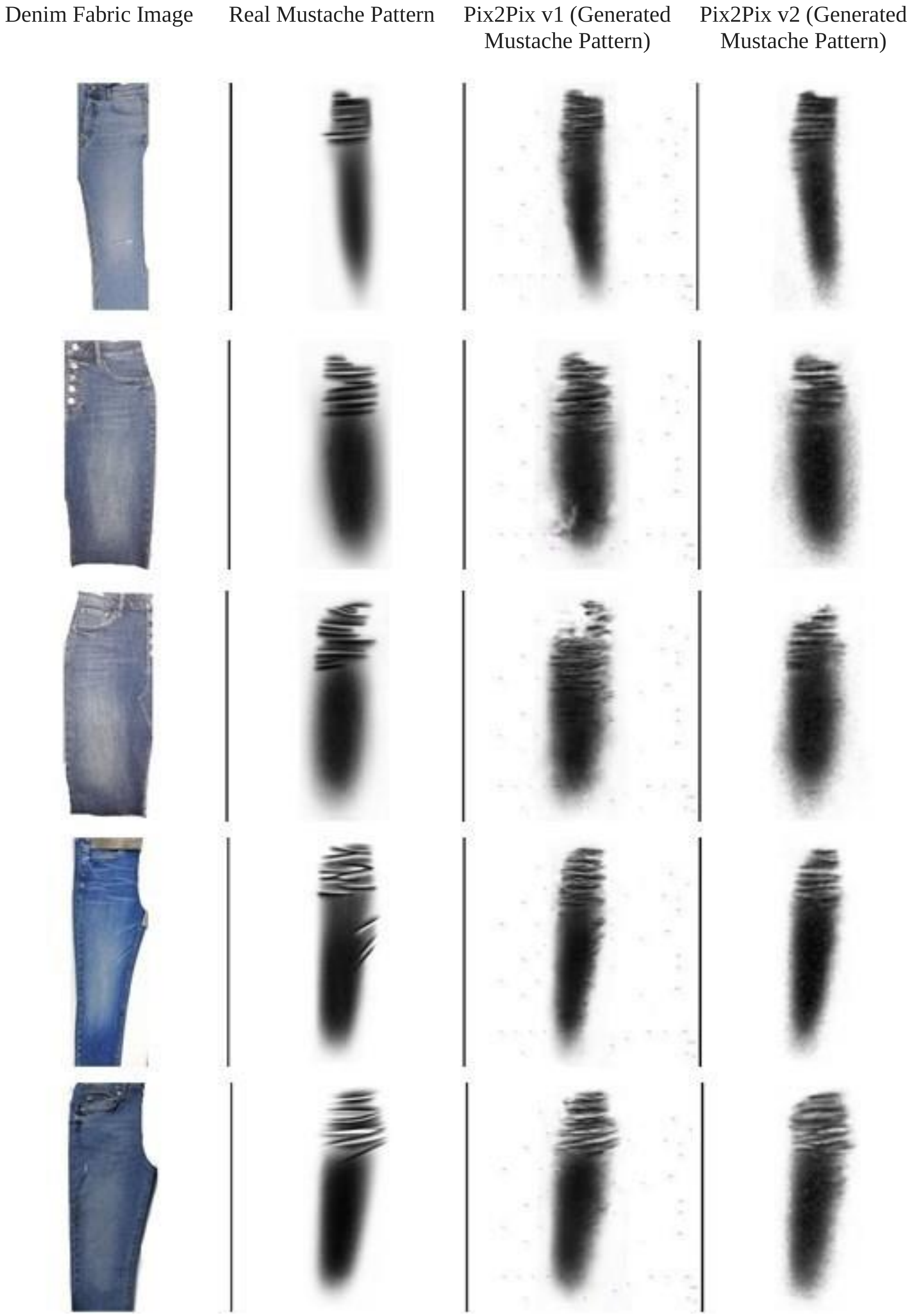

Figure 7. Test results of Denim2Mustache visual dataset.

References

[1] Zou, X., Wong, W. K., & Mo, D. (2018). Fashion Meets AI technology. In *International Conference on Artificial Intelligence on Textile and Apparel* (pp. 255-267). Springer, Cham.

[2] Jucienė, M., Urbelis, V., Juchnevičienė, Ž., & Čepukonė, L. (2014). The Effect of Laser Technological Parameters on The Color and Structure of Denim Fabric. *Textile Research Journal*, *84*(6), 662-670.

[3] Zhong, T., Dhandapani, R., Liang, D., Wang, J., Wolcott, M. P., Van Fossen, D., & Liu, H. (2020). Nanocellulose from Recycled Indigo-dyed Denim Fabric and Its Application in Composite Films. *Carbohydrate Polymers*, *240*, 116283.

[4] Golden Laser. (2021). Jeans Laser Engraving Machine. Retrieved from: https://www.goldenlaser.cc/jeans-laser-engraving-machine.html Accessed 21 June 2021

[5] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative Adversarial Nets. *Advances in neural information processing systems*, *27*.

[6] Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *arXiv preprint arXiv:1511.06434*.

[7] Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., ... & Shi, W. (2016). Photo-realistic Single Image Super-resolution Using a Generative Adversarial Network. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4681-4690).

[8] Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired Image-to-image Translation Using Cycle-consistent Adversarial Networks. In *Proceedings of the IEEE international conference on computer vision* (pp. 2223-2232).

[9] Karras, T., Aila, T., Laine, S., & Lehtinen, J. (2017). Progressive Growing of Gans for Improved Quality, Stability, and Variation. *arXiv preprint arXiv:1710.10196*.

[10] Park, T., Liu, M. Y., Wang, T. C., & Zhu, J. Y. (2019). Semantic Image Synthesis with Spatially-adaptive Normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 2337-2346).

[11] Wang, X., Xie, L., Dong, C., & Shan, Y. (2021). Real-esrgan: Training Real-world Blind Super-resolution with Pure Synthetic Data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 1905-1914).

[12] Huang, H., Yu, P. S., & Wang, C. (2018). An Introduction to Image Synthesis with Generative Adversarial Nets. *arXiv preprint arXiv:1803.04469*.

[13] Goodfellow, I. (2016). Nips 2016 tutorial: Generative Adversarial Networks. *arXiv preprint arXiv:1701.00160*.

[14] Atay, M. (2021). Generative Adversarial Networks. Retrieved from: https://colab.research.google.com/github/deeplearningturkiye/cekismeli-uretici-aglar-generative-adversarial-networks-gan/blob/master/gan-notebook-fresh.ipynb#scrollTo=toO2cXNQFYI5 Accessed 12 October 2021

[15] Mirza, M., & Osindero, S. (2014). Conditional Generative Adversarial Nets. *arXiv preprint arXiv:1411.1784*.

[16] Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired Image-to-image Translation Using Cycle-consistent Adversarial Networks. In *Proceedings of the IEEE international conference on computer vision* (pp. 2223-2232).

[17] Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-image Translation with Conditional Adversarial Networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1125-1134).

[18] Neurohiv. (2021). Pix2Pix – Image-to-Image Translation Network. Retrieved from: https://neurohive.io/en/popular-networks/pix2pix-image-to-image-translation/#:~:text=Pix2pix%20architecture,to%20get%20the%20output%20image. Accessed 2 October 2021

[19] Hu, J., Yu, W., & Yu, Z. (2017). Image-to-Image Translation with Conditional-GAN.

[20] Nilsson, J., & Akenine-Möller, T. (2020). Understanding Ssim. *arXiv preprint arXiv:2006.13846*.

[21] Ravuri, S., & Vinyals, O. (2019). Classification Accuracy Score for Conditional Generative Models. *arXiv preprint arXiv:1905.10887*.

[22] Willmott, C. J., & Matsuura, K. (2005). Advantages of The Mean Absolute Error (MAE) over The Root Mean Square Error (RMSE) in Assessing Average Model Performance. *Climate research*, *30*(1), 79-82.