

# Report on Digital Image Processing project

## AutoToon: Automatic Geometric Warping for Face Cartoon Generation

Le Thi Ha Thu-20224292

Hanoi University of Science and Technology (HUST)  
thu.lth224292@sis.hust.edu.vn

Nguyen Ngoc Thuy Linh-20224323

Hanoi University of Science and Technology (HUST)  
linh.nnt224323@sis.hust.edu.vn

### Abstract

AutoToon introduces an innovative method for generating cartoon-like faces through automatic geometric warping. This approach identifies and accentuates prominent facial features while preserving the subject's unique characteristics, resulting in visually engaging cartoon representations. The system combines face detection, landmark localization, and geometric transformations to achieve precise and controlled feature exaggeration. By leveraging both parametric and non-parametric warping techniques, AutoToon strikes a balance between artistic stylization and maintaining the subject's identity. The framework's modular design allows for adjustable levels of exaggeration and supports a variety of cartooning styles, making it versatile for creative applications. Evaluations highlight the system's effectiveness, with high user satisfaction and strong aesthetic appeal. AutoToon streamlines a traditionally manual process, offering significant potential for use in entertainment, social media, and personalized avatar creation.

### 1. Introduction

Early caricature generation methods relied on rule-based approaches, which were limited in diversity and effectiveness. Recent advances in deep learning, particularly in tasks like sketch synthesis, image translation, and style transfer, have reintroduced caricature generation as an image-to-image translation problem [5, 7]. However, while these systems achieve some level of geometric exaggeration and stylization, they often fall short in precisely targeting distinctive facial features or separating geometric warping from stylization, leading to less flexible and lower-quality results.

Geometric warping, the more challenging stage of caricature generation, has significant room for improvement.

While substantial progress has been made in general image stylization, face warping requires a higher degree of precision, as human perception is particularly sensitive to facial features. Poorly executed warping is more noticeable in photo-realistic images than in heavily stylized ones. Recognizing that there are already numerous high-quality stylization methods available, this work focuses on the harder problem: creating accurate geometric warps to produce high-quality, exaggerated face representations termed "cartoons."

In this project, we developed AutoToon, a supervised deep learning framework for geometric warping in caricature generation. By disentangling warping from stylization, AutoToon creates scalable, high-quality exaggerations that can pair with any artistic style. Using SENet and spatial transformer modules, the system learns from artist-generated warping fields and applies precise exaggerations. Additionally, we introduced the AutoToon dataset, a paired collection of facial portraits and warping fields, to advance research in cartoon generation. Through user studies and artist feedback, AutoToon demonstrated superior performance, producing visually appealing and effective exaggerations compared to existing methods.

### 2. Related work

#### 2.1. Learnt wrapping

Various methods have been developed to apply spatial transformations to images. Early techniques estimated global transformation parameters, while later approaches expanded to learning dense deformation fields across entire images [1]. For example, some methods have utilized dense flow estimation for tasks like gaze manipulation or removing geometric distortions in portrait images [2]. Others have employed spline interpolation on pre-detected landmarks to warp portraits while preserving identity [3]. Addi-

tional techniques have incorporated smoothness, local, and global alignment terms for tasks like parallax-tolerant image stitching [4]. Building on the success of these advancements, this work integrates dense flow estimation and a differentiable warping module to predict warping fields, which are then applied to create cartoons.

## 2.2. Caricature Generation

A key objective in caricature generation is to identify and exaggerate the distinct features of a given face. Traditional methods typically achieved this by highlighting differences from the average face, either through explicit landmark detection and warping or by using data-driven techniques to estimate unique facial features. Earlier approaches were mainly rule-based, which constrained the diversity of the generated caricatures. In more recent developments, deep learning methods have been employed. Modern caricature generation techniques are primarily data-driven. Some available datasets of annotated caricatures, such as Web-Caricature, contain 6042 caricatures and 5974 photographs from 252 distinct identities. However, the limited size of such datasets remains a significant challenge. As a result, much of the recent work in this area has drawn inspiration from generative image-to-image translation techniques, particularly those trained on unpaired images, with a focus on learning from unpaired portrait and caricature pairs.

## 3. Methodology

### 3.1. Dataset

A dataset of 101 portrait images featuring frontal-facing individuals (non-celebrities) was gathered from Flickr. The selected individuals represent a wide variety of age groups, genders, races, and facial shapes. These images were then transformed into caricatures, producing the ground-truth caricatures. The dataset was divided into 90 training images and 11 validation images. The test set, which does not have ground-truth labels, was collected from various subjects and public sources.

Additionally, the dataset includes estimated artist warping fields, denoted as  $F_{32} \in \mathbb{R}^{32 \times 32 \times 2}$ , which correspond to each artist’s caricature after bilinear upsampling to a size of  $H \times W \times 2$ . The choice of a  $32 \times 32$  spatial size for the warping field is discussed in the next section. To generate these fields, gradient descent optimization was performed on the warping field for each  $X_{toon}$  using  $L1$  loss through the differentiable Warping Module [2]. The optimization was aimed at minimizing the following expression:

$$\operatorname{argmin}_{F_{32}} \|X_{toon} - \text{Warp}(X_{in}, \text{Upsample}(F_{32}))\|_1 \quad (1)$$

### 3.2. Model Architecture

AutoToon, the method we propose for cartoon generation. The core of AutoToon’s exaggeration process consists of two key components: the Perceiver Network and the Warping Module. The Perceiver Network is based on a truncated version of the Squeeze-and-Excitation Network (SENet50), with weights pretrained on the VGGFace2 dataset due to its exceptional facial recognition performance. Specifically, we modify the network by retaining only the layers up to and including the second bottleneck block, followed by an adaptive average pooling layer that produces an output size of  $32 \times 32 \times 2$ . The truncation of the network helps reduce its capacity, preventing overfitting on the small dataset. The Perceiver Network takes the input image  $X_{in}$  and generates the warping field  $F_{32} \in \mathbb{R}^{32 \times 32 \times 2}$ , which is then bilinearly upsampled to obtain  $F$ , representing the per-pixel displacement. The Warping Module applies this warping field  $F$  to the input image  $X_{in}$  to produce the cartooned output  $X_{toon}$ . During inference, the warping field can also be scaled to adjust the intensity of the warp.

The decision to upsample the  $32 \times 32$  warping field was driven by two main factors. First, upsampling helps smooth the warps, which intuitively leads to smoother cartoons. Second, adhering to powers of 2, a  $64 \times 64$  warping field would have been too detailed, while a  $16 \times 16$  field resulted in less exaggerated cartoons (further details are available in the supplementary materials).

### 3.3. Loss Functions

We introduce three loss functions to train AutoToon: the reconstruction loss, artist warping loss, and smoothness regularization loss.

The reconstruction loss, denoted as  $L_{recon}$ , penalizes the  $L1$  distance between the generated cartoon  $X_{toon}$  and the artist-created cartoon  $X_{toon}$ . Along with supervising the model’s output, we also supervise the warping fields by comparing them with the artist’s warping fields [1]. The artist warping loss,  $L_{warp}$ , penalizes the  $L1$  distance between the artist’s warping field  $F_{32}$  (obtained using equation (1)) and the estimated warping field  $F_{32}$ .

Additionally, we incorporate a cosine similarity regularization loss,  $L_{reg}$ , to encourage smoothness in the warping field, reducing abrupt changes in contour. This is defined as:

$$L_{reg} = \sum_{i,j} \left( \frac{F_{ij} \cdot F_{ij+1}}{\|F_{ij}\| \|F_{ij+1}\|} \right) \quad (2)$$

where  $F_{ij} \cdot F_{ij+1}$  represents the dot product of the upsampled warping field  $F$  at pixel indices  $i, j$  and  $i, j + 1$ .

Thus, the total loss function used to train our model is:

$$L_{AutoToon} = \lambda_1 L_{recon} + \lambda_2 L_{warp} + \lambda_3 L_{reg} \quad (3)$$

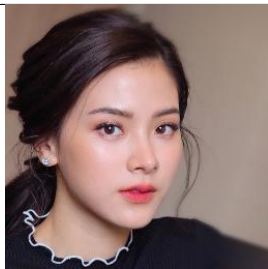
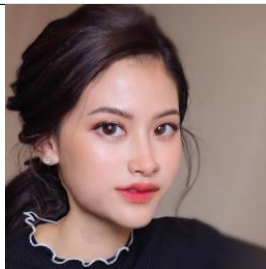


## 4. Experiment and Result

### 4.1. Training Details

We use the Adam optimizer with parameters  $\beta_1 = 0.5$  and  $\beta_2 = 0.999$ , and a learning rate decay factor of 0.95. The batch size is fixed at 16, where each mini-batch contains a randomly selected and aligned input-cartoon pair along with the corresponding artist warp. For data augmentation, we apply two methods: random horizontal flipping and color jittering. The color jittering involves random adjustments to brightness, contrast, and saturation (each uniformly sampled from the range  $[0.9, 1.1]$ ), as well as hue shifts (uniformly sampled from  $[-0.05, 0.05]$ , following PyTorch’s color jitter API). The hyperparameters are empirically set as  $\lambda_1 = 1$ ,  $\lambda_2 = 0.7$ , and  $\lambda_3 = 1e-6$ .

### 4.2. Good test results

Table 1. Examples of good tests

Original	AutoToon
	
	





### 4.3. Bad test results

Images with solid, high-contrast backgrounds or those not normalized can show unexpected exaggerations when processed by the AutoToon model. Pre-processing steps like background normalization can help achieve consistent and visually pleasing results.

When a person’s face is partially obscured, tilted, or involves two individuals with one person’s inner eyes not recognized, the AutoToon model selectively exaggerates only the recognized parts of the face. This results in a distinctive half-exaggerated aesthetic. Pre-processing steps to enhance face recognition or handle obscured or tilted faces can help achieve a more comprehensive exaggeration of facial features.

The attempts to exaggerate these images failed during preprocessing as the landmark detectors couldn’t recognize any faces. This hindered normalization, highlighting the need for adjustments. Exploring alternative landmark detection algorithms or fine-tuning parameters may enhance face recognition in challenging conditions, improving the preprocessing workflow.

Table 2. Examples of bad tests

Original	AutoToon	Problem
		Half exaggeration
		Unusual exaggeration

## 5. Evaluation

### 5.1. MSE (Mean Squared Error): 479

The formula for calculating Mean Squared Error (MSE) is:

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (X_i - Y_i)^2 \quad (4)$$

where:

- $N$  is the number of pixels in the image.
- $X_i$  is the pixel value at position  $i$  in the ground truth image.
- $Y_i$  is the pixel value at position  $i$  in the output image.

MSE calculates the average squared error between the pixels of the ground truth image and the output image.

A value of 479 indicates a significant difference between the two images at the pixel level. This value is relatively high, suggesting that the output image from AutoToon has undergone substantial transformations compared to the original.

### 5.2. SSIM (Structural Similarity Index): 0.86

The formula for calculating the Structural Similarity Index (SSIM) is:

$$\text{SSIM}(x, y) = \frac{(2xy + C_1)(2\sigma_{xy} + C_2)}{(x^2 + y^2 + C_1)(x^2 + y^2 + C_2)} \quad (5)$$

where:

- $x$  and  $y$  are the mean values of images  $x$  and  $y$ .
- $x^2$  and  $y^2$  are the variances of images  $x$  and  $y$ .
- $\sigma_{xy}$  is the covariance between images  $x$  and  $y$ .

- $C_1$  and  $C_2$  are constants used to stabilize the formula when dividing by small numbers.

SSIM evaluates the structural similarity between two images, with values closer to 1 indicating higher similarity, meaning the output image is more similar to the ground truth image. An SSIM value of 0.86 is quite good and suggests that, despite the noticeable pixel-level differences, the structural integrity and overall visual characteristics of the original image have been well-preserved. This result is a strong indicator that AutoToon successfully maintains the essential features of the input, such as facial shapes and contours, while applying its stylized effects.

### 5.3. PSNR (Peak Signal-to-Noise Ratio): 21.32 dB

The formula for calculating Peak Signal-to-Noise Ratio (PSNR) is:

$$\text{PSNR} = 10 \cdot \log_{10} \left( \frac{\text{MAX}_I^2}{\text{MSE}} \right) \quad (6)$$

where:

- $\text{MAX}_I^2$  is the maximum value of a pixel in the image.
- MSE is the Mean Squared Error between the ground truth image and the output image.

A PSNR of 21 dB falls into the "moderate" range. While it is not exceptionally high, it is not unexpected for a system like AutoToon. This model's goal is not to preserve the exact pixel details of the input image but rather to create a stylistically altered version. Therefore, the moderate PSNR reflects the deliberate introduction of differences due to the cartoonization process. This result aligns with the nature of AutoToon, as the focus is on stylization rather than pixel-perfect fidelity..

## 6. Conclusion

In this project, we successfully implemented the AutoToon system, which is the first supervised deep learning method for cartoonization. It achieves high-quality warping while preserving facial details. By separating the warping process from stylization, AutoToon provides flexibility and retains detail, making it better than existing methods at exaggerating facial features. Using a dataset of 101 image pairs and comparing with previous techniques on key criteria, AutoToon has shown strong potential for real-world applications.

However, the system still has some limitations. In some cases, the output has little or half exaggeration, making it look almost identical to the input image. This can happen even when there are no major differences in the input or due to adjustments in certain model parameters, which may reduce the output quality.

In the future, improvements could focus on making the warping smoother, better preserving the subject's identity, and adapting the model to various artistic styles using few-shot learning.

## 7. References

1. DeepWarp: Photorealistic Image Resynthesis for Gaze Manipulation (<https://arxiv.org/pdf/1607.07215>)
2. Learning Perspective Undistortion of Portraits ([https://openaccess.thecvf.com/content\\_ICCV\\_2019/papers/Zhao\\_Learning\\_Perspective\\_Undistortion\\_of\\_Portraits\\_ICCV\\_2019\\_paper.pdf](https://openaccess.thecvf.com/content_ICCV_2019/papers/Zhao_Learning_Perspective_Undistortion_of_Portraits_ICCV_2019_paper.pdf))
3. Synthesizing Normalized Faces from Facial Identity Features (<https://arxiv.org/pdf/1701.04851>)
4. Parallax-tolerant Image Stitching ([https://openaccess.thecvf.com/content\\_cvpr\\_2014/papers/Zhang\\_Parallax-tolerant\\_Image\\_Stitching\\_2014\\_CVPR\\_paper.pdf](https://openaccess.thecvf.com/content_cvpr_2014/papers/Zhang_Parallax-tolerant_Image_Stitching_2014_CVPR_paper.pdf))
5. CariGANs: Unpaired Photo-to-Caricature Translation (<https://arxiv.org/pdf/1811.00222>)
6. WarpGAN: Automatic Caricature Generation (<https://arxiv.org/pdf/1811.10100>)