CS230

Origami Model Generation with Neural Style Transfer and CycleGAN

Yuina Iseki
Department of Computer Science
Stanford University
yuina@stanford.edu

Changju Yuan
Department of Civil and Environmental Engineering
Stanford University
ycj2003@stanford.edu

Antra Nakhasi
Department of Management Science and Engineering
Stanford University
anakhasi@stanford.edu

Abstract

This project transforms animal images into origami representations using Neural Style Transfer (NST) and CycleGAN, motivated by applications in bio-inspired deployable structure design. We implement and evaluate three architectures: vanilla NST with custom VGG-19 layer configurations, feed-forward NST with stylized residuals, and CycleGAN with encoder-residual-decoder generators. Experimental results demonstrate that vanilla NST with position-matched and segmented image pairs outperform other approaches when using configurations specifically designed for planar surfaces and geometric emphasis. Feed-forward NST models show stable training dynamics but remain limited to texture-level transfer, while CycleGAN struggles with the unpaired nature of the dataset and the fundamental challenge of imposing origami's geometric constraints without explicit structural supervision. Results show that successful origami-style translation requires position-matched animal-to-origami image pairs, careful image preprocessing, and architecture designs that explicitly encode geometric priors, suggesting that future work should combine hybrid approaches with geometric constraints across diverse animal anatomies.

1 Introduction

In this project, we aim to transform real-world animal images into origami model images by experimenting with various combinations of style transfer algorithms, hyperparameter tuning, and loss functions. We focus our quantitative experiments on butterfly images due to their bilateral symmetric structure that enables easier pose matching, and wing structure that exhibits the planar surfaces central to origami geometry. The input to our algorithm is a real-world butterfly image; we then used Neural Style Transfer(NST) and Cycle-Consistent Generative Adversarial Networks(CycleGAN) to generate an origami model image corresponding to the input image. Mathematical principles in origami are driving innovations in deployable systems across robotics, architecture, biomedical devices, and aerospace. By exploring an AI tool that interprets natural forms through an origami lens, we hope to assist engineers in designing foldable, compact, and rapidly deployable structures for various real-world applications. Our implementation and datasets are publicly available.[1]

2 Related work

Recent work on style transfer and unpaired translation provides the main technical foundation for our approach. Classical neural style transfer by Gatys et al. formulates style using Gram-matrix statistics over deep features, producing high-fidelity textures but requiring slow per-image optimization and offering limited control over geometric structure [2]. Johnson et al. introduced a feed-forward perceptual-loss network that enables real-time stylization, but fixes a single style per model and similarly struggles to enforce structural constraints [3].

For unpaired domain translation, CycleGAN introduced cycle-consistency to learn mappings between two domains without paired data and remains a widely used baseline for tasks involving large appearance shifts [4]. Extensions such as MUNIT and DRIT disentangle content and style into separate latent spaces, enabling multi-modal sampling but still focusing mainly on texture rather than planar or faceted geometry [5, 6]. Attention-based formulations like AttentionGAN improve structure

[1] See Appendix A for code and dataset links.

preservation by learning spatial masks that guide where the generator should modify the image, a clever mechanism for retaining local semantic boundaries under strong style transformations [7].

In parallel, computer-graphics research on origami rendering and modeling provides explicit geometric priors: Furuta et al. simulate paper reflectance and crease shading [8], while Mitani and Igarashi model fold lines and planar facets to synthesize realistic paper sculptures [9]. These methods offer precise geometric fidelity but require hand-engineered models rather than learned image-to-image mappings.

Our work lies between these directions. Like CycleGAN and its extensions, we learn from unpaired data; however, our target domain (origami) requires structural consistency that classical NST and feed-forward NST do not model. And unlike graphics-based origami approaches, we aim to learn the mapping automatically from images rather than manually designing folds. Since realistic origami generation is still mostly done by hand, this problem remains relatively unexplored

3 Dataset

The project uses two unpaired visual domains: real world animal photographs sourced from ImageNet[1] and origami paper models sourced from two publicly available Kaggle datasets.[2] After label normalization, duplicate removal, RGB conversion, and resizing, the combined dataset contains 61,201 images (56,814 animal images and 4,387 origami images) spanning 106 species categories and split 80/10/10 across training, validation, and testing. Because no one-to-one correspondence exists between animal and origami images, we applied extensive preprocessing and augmentation to improve domain coverage. Animal images were segmented using a YOLOv8 model to isolate foreground subjects, while the smaller origami domain underwent targeted augmentation, including horizontal flips, small rotations (up to $\pm 20^{\circ}$), random resized crops (scale 0.8–1.0), color jitter, and mild perspective distortion. All images were resized to either 256×256 or 512×512 for training and normalized using ImageNet mean–std statistics for NST or scaled to $[-1, 1]$ for CycleGAN.

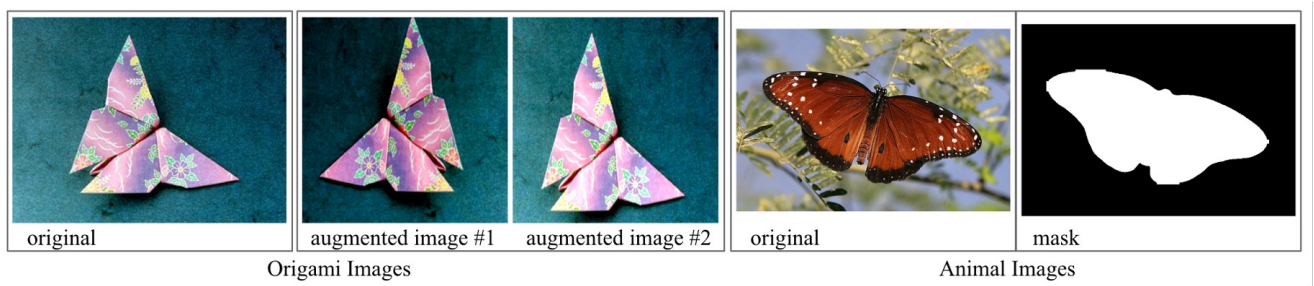

Figure 1: Examples of preprocessing: augmented origami images (left) and an animal image with its segmentation mask (right)

4 Methods

4.1 Vanilla NST

Vanilla NST uses a pre-trained VGG-19 model to extract content and style features from multiple convolutional layers at multiple scales. The loss function is:

$$\mathcal{L}_{total} = \alpha \cdot \mathcal{L}_{content} + \beta \cdot \mathcal{L}_{style} + \gamma \cdot \mathcal{L}_{tv} \tag{1}$$

where the content and style loss are equivalent to those defined in Gatys et al. [2] We optimize this loss with respect to pixel values using Adam ($\alpha_{lr} = 0.003$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, $\gamma = 0$), initializing from the content image. Beyond the original Gatys configuration, we designed custom layer selections emphasizing geometric features: `planar_surfaces` uses mid-to-deep layers (conv3_1, conv4_1) to capture flat regions, `edge_heavy` uses early layers (conv1_1, conv2_1) with high weights for sharp folds, and `geometric_emphasis` focuses on mid-level features (conv2_1, conv3_1, conv4_1) that encode angular structures.[3]

4.2 Feed-Forward NST

Feed-forward NST models train a Transformer Network that produces a stylized output in a single forward pass using the perceptual content, style, and total-variation losses originally introduced by Gatys et al. [2] These losses are used exactly as defined in the original paper; the only difference is that, following the feed-forward approach of Johnson et al. [3], they are

[2]Additional dataset sources and details are provided in Appendix A.
[3]Full configuration details in Appendix B.

not optimized over the output image itself but instead serve as supervision for learning the parameters of the Transformer Network. The overall training objective is same as 1

The Transformer architecture is shared across all configurations and differs only in the type of residual block used:
Baseline Model (3 convolutional layers $\rightarrow$ 5 standard residual blocks, each containing two 3×3 convolutions with InstanceNorm and ReLU plus a skip connection $\rightarrow$ 2 upsampling layers) and
Stylized-residual Model (each residual block has an additional 1×1 convolution and a sigmoid gating module).

Three training configurations were tried,[4] each differing only in how their perceptual losses are defined while keeping the architecture fixed. The Uniform Style Model follows the classical feed-forward NST setup, using a standard 3-channel RGB input, the full Gatys style-layer set, and moderate style weights (1.00.1) with a learning rate of 1×10^{-3}. The Mask-Guided Style Model introduces partial supervision: the network receives a 4-channel input (RGB + mask), uses a lower learning rate of 1×10^{-4}, applies data augmentation, and relies on a single pre-averaged Gram matrix to stabilize training. The Geometry-Aware Style Model retains this supervised structure and 4-channel input but replaces the full Gatys layer set with a mid-level VGG selection (conv2_1–conv4_1) and increases style weights ([1.5, 1.5, 1.0]) while reducing the TV weight to 1×10^{-6} to emphasize origami-like edges, folds, and planar surfaces. The behavioral differences observed in later sections arise solely from these changes in perceptual loss design.

4.3 CycleGAN

Vanilla CycleGAN Model: The generator follows an encoder-residual block-decoder architecture: three down-sampling convolutional layers, nine residual blocks(each with two 3×3 convolutions and a skip connection), and two up-sampling ConvTranspose layers ending with a `tanh` activation. The discriminator adopts a 70×70 receptive field, with 4 convolutional layers and LeakyReLU(coefficient$= 0.2$) activation. Its output is a 30×30 map of real/fake probabilities, where the average determines the discriminator's confidence. The total objective combines adversarial, cycle-consistency, and identity terms:

$$\mathcal{L} = \mathcal{L}_{GAN}(G, D_Y) + \mathcal{L}_{GAN}(F, D_X) + \lambda_{cyc}\mathcal{L}_{cyc}(G, F) + \lambda_{id}\mathcal{L}_{id}(G, F) \tag{2}$$

where $\lambda_{cyc} = 10$ and $\lambda_{id} = 0.5\lambda_{cyc}$ [3]. Adversarial losses encourage stylistic realism, while the cycle term enforces structure preservation through $F(G(x)) \approx x$. The model is optimized with Adam(lr $= 2 \times 10^{-4}$ for F and G, and lr $= 1 \times 10^{-4}$ for D_X and D_Y, with batch size 4. Training uses alternating optimization of G/F and D_X/D_Y with TensorFlows automatic differentiation.

CycleGAN Model with Perceptual Loss: The original CycleGAN architecture is retained, but the cycle-consistency objective is augmented with a perceptual feature constraint computed using VGG19. Specifically, in addition to pixel-wise reconstruction, feature maps extracted from the `block3_conv3` layer of a frozen ImageNet-pretrained VGG19 network are used to enforce high-level structural fidelity. This allows the generator to match semantic contours and textures rather than merely minimizing low-level L_1 distance. Also, in contrast to the vanilla CycleGAN objective, where both generators are optimized through a single combined loss as Eq. 2, in this model we decompose the cycle-consistency loss into a pixel term and a VGG19-based perceptual term. The overall generator objective can then be written as

$$\mathcal{L} = \mathcal{L}_{GAN}(G, D_Y) + \mathcal{L}_{GAN}(F, D_X) + \lambda_{cyc}\,\mathcal{L}_{cyc}^{total}(G, F) + \lambda_{id}\,\mathcal{L}_{id}(G, F), \tag{3}$$

where $\mathcal{L}_{cyc}^{total}(G, F)$, includes both pixel-level and perceptual cycle consistency terms. [5]

In addition, we explicitly separate training signals to avoid inverse-gradient interference during the training cycle. The objectives for the forward and backward mappings are written as:

$$\mathcal{L}_G = \mathcal{L}_{GAN}(G, D_Y) + \lambda_{cyc}\mathcal{L}_{cyc}(G) + \lambda_{id}\mathcal{L}_{id}(G) \qquad \text{and} \qquad \mathcal{L}_F = \mathcal{L}_{GAN}(F, D_X) + \lambda_{cyc}\mathcal{L}_{cyc}(F) + \lambda_{id}\mathcal{L}_{id}(F) \tag{4}$$

This decomposition prevents gradient coupling between two [5], ensuring that updates to G do not implicitly perturb the training efficiency of F, and vice versa. In practice, we observe improved adversarial stability and reduced oscillation in cycle reconstruction once the two update paths are decoupled.

5 Experiments, Results and Discussion

We evaluate each configuration by comparing generated images to their content inputs using SSIM (Structural Similarity Index) for perceptual quality and PSNR (Peak Signal-to-Noise Ratio in dB) for pixel-level fidelity.

[4] Detailed hyperparameters for all feed-forward NST variants and rationale behind choosing certain layers are provided in Appendix C.
[5] The complete equations for total cycle loss and separate equations after decoupling are in Appendix D

5.1 Vanilla NST

The results demonstrate that image preprocessing significantly impacts NST performance, with position-matched consistently outperforming unmatched base images.[6] Notably, the planar_surfaces configuration excelled on matched-segmented images, achieving the highest PSNR (25.77 dB) and exceptional SSIM (0.677 on matched images), indicating superior preservation of origami's characteristic flat, angular surfaces when background noise is removed. These findings suggest that the optimal NST configuration depends heavily on the quality and availablity of position-matched image-to-image dataset and its preprocessing strategy: it enables better style transfer by reducing background interference and ensuring spatial correspondence between content and style images.

Table 1: Comparison of best performing Vanilla NST configurations

Pair of Images	**Configuration**	**SSIM ↑**	**PSNR ↑**	**Gram Dist. ↓**	**Total Loss ↓**
Matched	minimal_fast	0.54931	21.20848	0.207041×10^6	$\mathbf{0.43435}\times10^6$
Matched and Segmented	planar_surfaces	0.367293	**25.76927**	0.482482×10^6	1.874488×10^6
Matched	edge_heavy	0.509557	19.63971	$\mathbf{0.1155645}\times10^6$	1.874488×10^6
Matched	planar_surfaces	**0.677251**	23.93335	0.387716×10^6	1.129015×10^6

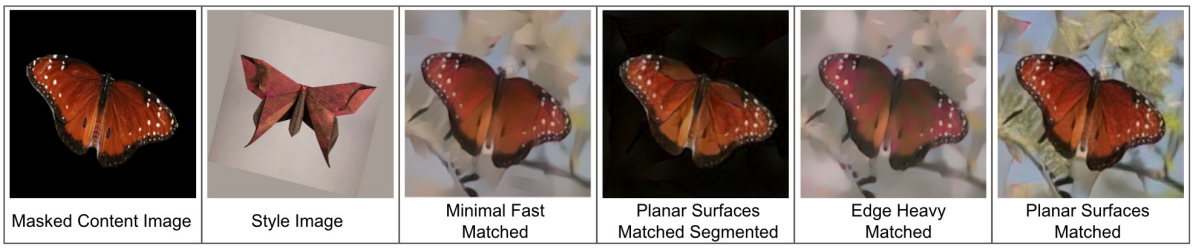

Figure 2: Comparison of best performing Vanilla NST configurations.

5.2 Feed-Forward NST

We evaluated three feed-forward NST systems: the *Uniform Style Model*, the *Mask-Guided Style Model*, and the *Geometry-Aware Style Model*, each producing a baseline and a stylized output version. Figure 3 shows the six generated outputs (baseline and stylized for each model) on the same butterfly input image.

In the baseline version of the Uniform Style Model, the output looks washed out but few splotches of colors from the origami paper appear throughout the image, causing the output to look washed out. Only faint structural details of the butterfly, such as light wing outlines and minimal shape definition, remain visible. The stylized version continues this behavior but limits it mostly into the butterfly region. These effects correspond to its low SSIM values (0.521 baseline, 0.586 stylized) and extremely high Gram distance (7.64×10^5), indicating that the model alters the entire image without preserving subject structure.

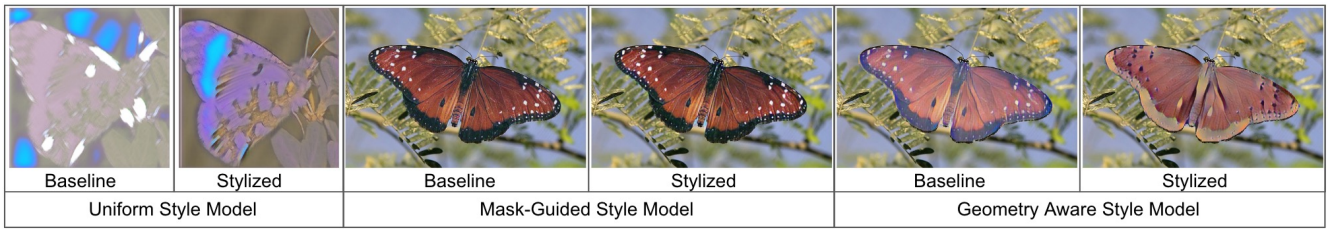

Figure 3: Outputs of all feed-forward NST models (baseline and stylized)

The Mask-Guided Style Model behaves very differently. The baseline output remains close to the original butterfly, maintaining clear edges, and changing only subtle parts of the animal. The stylized version looks nearly identical to the baseline, with only a very slight increases in texture. This stability is reflected in the highest scores among all models: SSIM reaches 0.864 (baseline) and 0.854 (stylized), PSNR reaches $17\ dB$, and the Gram distance drops to 1.83×10^{-6}. This model indicated that mask ensures strong structure preservation and consistent localization of style.

The Geometry-Aware Style Model produces the most pronounced artistic transformation amongst the feed forward NSTs. Its baseline output shows light stylization limited to the butterfly, with faint texture . Its stylized output, however, becomes slightly angular and more textured, and flattened shading resembling origami folds. These visual changes explain its lower

[6]Full comparison of final images provided in Appendix E.

measurements (SSIM of 0.516 baseline and -0.059 stylized, PSNR around $10dB$). Although these values are low, they show that the model is deliberately balancing geometric stylization with how much of the original image it keeps.

The Uniform Style Model highlights the limitations of unconstrained NST, the Mask-Guided Style Model effectively preserves the animal and therefore verifies the need for segmentation, and the Geometry-Aware Style Model demonstrates the best style transfer out of the 3 feed forward NST. Yet all three feed-forward NST models perform worse than Vanilla NST, for example, the best Vanilla configuration reaches SSIM $= 0.6773$ and PSNR $= 23.93$, noticeably higher than even the strongest feed-forward result (the Mask-Guided baseline achieves SSIM $= 0.864$ but only PSNR $= 17.89$). This gap shows that, despite localized improvements, the feed-forward systems cannot match Vanilla NST's overall balance of structure preservation and texture accuracy.

Table 2: Results for all NST models (baseline and stylized). Best values are in bold.

Model	Variant	PSNR	SSIM	Gram	Loss
Uniform Style Model	Baseline	18.92	0.521	7.64×10^5	2.16×10^{12}
	Stylized	9.60	0.586	4.06×10^{-8}	0.45
Mask-Guided Style Model	Baseline	**17.89**	**0.864**	1.83×10^{-6}	**0.34**
	Stylized	**17.34**	**0.854**	1.93×10^{-6}	**0.50**
Geometry-Aware Style Model	Baseline	11.56	0.516	1.04×10^{-6}	2.85
	Stylized	10.02	-0.059	7.86×10^{-6}	3.06

5.3 CycleGAN

In addition to the baseline translation setup, a fourth input channel was introduced to supply segmentation masks alongside the RGB image. The evaluation indicates that mask-segmented CycleGAN maintains reasonable image translation fidelity while showing clear structural preservation under segmentation-guided training. The SSIM (0.49) and PSNR (13.9 dB) reflect moderate pixel-level alignment between generated origami imagery and ground-truth targets, while the extremely high Gram distance (1.18×10^8) suggests that the model reproduces global geometric structure without fully capturing fine-grained style statistics in deep feature space.

Introducing perceptual loss shifts the model toward stylistic studying rather than pixel-level accuracy. This version displays lower SSIM (0.417) and PSNR (9.93), along with a larger Gram distance and value of loss function, indicating a measurable drop in pixel-level fidelity. As the result image shows, the perceptual objective successfully intensifies texture, hue, and artistic domain cues, but does so by relaxing structural constraints that previously stabilized shape and segmentation boundaries.

Table 3: Results for masked Vanilla vs. Perceptual CycleGAN models. Best values are in bold.

Model Variant	PSNR ↑	SSIM ↑	Gram Dist. ↓	Total Loss ↓
Masked Vanilla	**13.90**	**0.499**	**1.18**$\times 10^8$	**4.77**
Perceptual CycleGAN	9.93	0.417	2.70×10^8	9.73

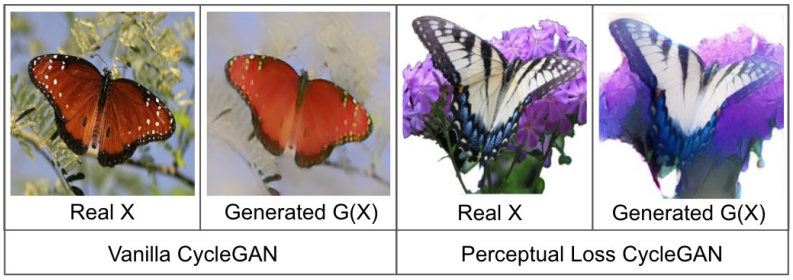

Figure 4: Outputs of both CycleGAN models

Across results from two models, stronger stylization does not automatically improve perceptual coherence. The more aggressively the model pursues domain-specific texture, the more it compromises structural fidelity and pixel-level similarity. In other words, for domain transformation tasks, the stylistic gain visible in the perceptual model comes with a cost in geometric clarity and reconstruction quality. This suggests that model choice is a trade-off depending on desired emphasis: realism and contour accuracy for vanilla model, or artistic abstraction for perceptual model.

6 Conclusion/Future Work

This work demonstrated that effective animal-to-origami translation requires explicit geometric reasoning, well beyond the capacity of standard image-to-image models. Of the three methods investigated here, Vanilla NST resulted in the best perfor-

mance when paired with position-matched and segmented inputs. Feed-forward NST remained constrained by limited texture transfer, while CycleGAN failed to learn proper geometric transformations under unpaired training.

These findings stress the importance of data preprocessing, particularly background removal and pose alignment,suggesting that the origami translation problem is fundamentally a structured domain gap rather than just an artistic/texture transfer task.

Our future work should be guided in three directions. First, hybrid architectures, which encode geometric constraints (such as planarity losses, crease-aware edge detection). Second, collecting a paired dataset with pose-aligned animal and origami images. Third, evaluation incorporating fold-quality metrics extending beyond pixel similarity.

With such advances, learning-based translation systems could start to support designers and engineers in automatically generating foldable, deployable structural concepts inspired by natural forms that bridge computational design, origami engineering, and computer vision.

7 Contributions

The individual contributions are as follows:

- Yuina Iseki: Implemented and evaluated vanilla NST with custom VGG-19 layer configurations optimized for geometric structure preservation. Designed and tested eight configuration variants across matched, segmented, and base image pairs. Conducted quantitative analysis using SSIM, PSNR, and Gram distance metrics. Contributed to abstract, introduction, conclusion, and vanilla NST section of paper.
- Changju Yuan: Implemented and trained the CycleGAN baseline and perceptual loss architectures with encoder-residual-decoder generators. Conducted hyperparameter tuning and epoch-wise evaluation. Developed the CycleGAN variant with perceptual loss augmentation and decoupled generator optimization. Contributed to CycleGAN section of paper.
- Antra Nakhasi: Developed and trained the feed-forward NST architectures, including baseline and stylized-residual variants. Conducted hyperparameter tuning and epoch-wise evaluation across all feed-forward NST variants. Managed dataset preprocessing, augmentation pipeline, and segmentation using YOLO model. Contributed to related works, dataset description, and feed-forward NST section of paper. Additionally, configured and managed the AWS compute infrastructure, including EC2, SageMaker, and S3 storage used throughout model training and experimentation.

References

[1] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition (pp. 248-255). Ieee.

[2] Gatys, L. A., Ecker, A. S., Bethge, M. (2015). A neural algorithm of artistic style. arXiv preprint arXiv:1508.06576.

[3] Johnson, J., Alahi, A., Fei-Fei, L. (2016, September). Perceptual losses for real-time style transfer and super-resolution. In European conference on computer vision (pp. 694-711). Cham: Springer International Publishing.

[4] 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).

[5] Huang, X., Liu, M. Y., Belongie, S., Kautz, J. (2018). Multimodal unsupervised image-to-image translation. In Proceedings of the European conference on computer vision (ECCV) (pp. 172-189).

[6] Lee, H.-Y., Huang, J.-B., Singh, M., Yang, M.-H. (2018). Diverse image-to-image translation via disentangled representations (DRIT). ECCV.

[7] Alami Mejjati, Y., Richardt, C., Tompkin, J., Cosker, D., Kim, K. I. (2018). Unsupervised attention-guided image-to-image translation. Advances in neural information processing systems, 31.

[8] Mitani, J. (2009). A design method for 3D origami based on rotational sweep. Computer-Aided Design and Applications, 6(1), 69-79.

Appendix

A Code Repository and Dataset

The implementation and training scripts are available at:
`https://github.com/antranakhasi/Origami-Model-using-CycleGAN`

The full dataset used can be downloaded at:
`https://drive.google.com/file/d/1HFv68vV3LQNo930ssLcYwW-RSdd0TJf_/view?usp=sharing`

Further Details on Dataset The animal domain is sourced from ImageNet, with images filtered by WordNet synsets to match the same 106 species categories used in the origami domain. The origami images come from two publicly available Kaggle datasets:

1. **Origami Models:**compiles origami images from community repositories including OriSet, OriWiki, and Gilad's Origami Page. (`https://www.kaggle.com/datasets/karthikssalian/origami-models`)
2. **Origami Works of Some Origamists:** aggregates photographs contributed by multiple origami artists, spanning diverse folding styles, materials, and crease patterns.
(`https://www.kaggle.com/datasets/caokhoihuynh/orgami-works-of-some-origamists`)

B Vanilla NST Metrics

Table 4: Vanilla NST Layer Configurations

Configuration	Content Layer	Style Layers	Style Weights	Rationale
`gatys`	conv4_2	conv1_1, conv2_1, conv3_1, conv4_1, conv5_1	1.0, 0.8, 0.5, 0.3, 0.1	Baseline configuration with proven performance; balanced multi-scale style transfer across all VGG19 layers
`geometric emphasis`	conv4_2	conv2_1, conv3_1, conv4_1	1.5, 1.5, 1.0	Focus on mid-level layers that capture geometric patterns and edges, ideal for origami's angular structures and sharp folds
`edge heavy`	conv4_2	conv1_1, conv2_1	2.0, 1.5	Early layers only with high weights to maximize sharp fold and crisp edge detection in the style transfer
`planar surfaces`	conv4_2	conv3_1, conv4_1	1.5, 1.5	Mid-to-deep layers that capture flat regions and smooth surfaces, reflecting paper's planar nature
`equal weights`	conv4_2	conv1_1, conv2_1, conv3_1, conv4_1, conv5_1	1.0, 1.0, 1.0, 1.0, 1.0	Removes hierarchical bias to let all scales contribute equally; unbiased baseline for comparison
`high detail content`	conv3_1	conv1_1, conv2_1, conv3_1, conv4_1	1.0, 1.0, 0.8, 0.5	Earlier content layer preserves fine details and structural features, potentially maintaining animal characteristics more clearly
`minimal fast`	conv4_2	conv2_1, conv3_1	1.0, 1.0	Minimal layer selection for faster computation; serves as efficient baseline for testing and iteration
`texture focused`	conv4_2	conv1_1, conv1_2, conv2_1, conv2_2	1.2, 1.0, 1.2, 1.0	Multiple early layers emphasize fine texture details, capturing paper texture and surface characteristics

C Feed-Forward NST Hyperparameters

Table 5: Hyperparameters used for each feed-forward NST model

Hyperparameter	Uniform	Mask-Guided	Geometry-Aware
Input channels	3 (RGB)	4 (RGB + mask)	4 (RGB + mask)
Learning rate α	1×10^{-3}	1×10^{-4}	1×10^{-4}
Batch size	6	4	4
Epochs	12	12	12
Style layers	Gatys	Gatys	conv2_1–conv4_1
Style weights β	[1.0–0.1]	[1.0–0.1]	[1.5, 1.5, 1.0]
TV weight γ	default	default	1×10^{-6}

Table 6: Geometry-Aware NST Layer Configuration

Model	Content Layer	Style Layers	Style Weights	Rationale
Geometry-Aware	conv4_2	conv2_1, conv3_1, conv4_1	1.5, 1.5, 1.0	captures geometric patterns: angular folds, planar surfaces, and subtle crease structure. Higher weights emphasize edges and shape abstraction rather than global color transfer.

D CycleGAN Equations

The pixel-cycle loss is

$$\mathcal{L}_{cyc}^{pix}(G,F) = \mathbb{E}_x\Big[\|F(G(x)) - x\|_1\Big] + \mathbb{E}_y\Big[\|G(F(y)) - y\|_1\Big], \tag{5}$$

and the perceptual-cycle loss is

$$\mathcal{L}_{cyc}^{perc}(G,F) = \mathbb{E}_x\Big[\|\phi(F(G(x))) - \phi(x)\|_1\Big] + \mathbb{E}_y\Big[\|\phi(G(F(y))) - \phi(y)\|_1\Big]. \tag{6}$$

where α controls the strength of perceptual supervision, we take its value 0.02 here to balance time consuming and model performance. The total cycle loss is:

$$\mathcal{L}_{cyc}^{total}(G,F) = \mathcal{L}_{cyc}^{pix}(G,F) + \alpha\,\mathcal{L}_{cyc}^{perc}(G,F), \tag{7}$$

After decoupling for training, two loss are

$$\mathcal{L}_{cyc}^{G} = \mathbb{E}_x\Big[\|F(G(x)) - x\|_1\Big] + \alpha\,\mathbb{E}_x\Big[\|\phi(F(G(x))) - \phi(x)\|_1\Big]. \tag{8}$$

$$\mathcal{L}_{cyc}^{F} = \mathbb{E}_y\Big[\|G(F(y)) - y\|_1\Big] + \alpha\,\mathbb{E}_y\Big[\|\phi(G(F(y))) - \phi(y)\|_1\Big]. \tag{9}$$

E Vanilla NST Complete Results

Figure 5: Full Comparison of vanilla NST outputs.