References

[1] Liu, Y., Qin, Z., Wan, T. and Luo, Z. (2018). Auto-painter: Cartoon image generation from sketch by using conditional Wasserstein generative adversarial networks. *Neurocomputing*, 311, pp.78–87. doi:https://doi.org/10.1016/j.neucom.2018.05.045.

[2] Chen, S., Zhang, J., Gao, L., He, Y., Xia, S., Shi, M. and Zhang, F.-L. (2022). Active Colorization for Cartoon Line Drawings. IEEE Transactions on Visualization and Computer Graphics, 28(2), pp.1198–1208. doi:https://doi.org/10.1109/tvcg.2020.3009949.

[3] Software Informer. (2023). RETAS STUDIO PaintMan. Get the software safely and easily. [online] Available at: https://retas-studio-paintman.software.informer.com/6.5 [Accessed 27 Apr. 2024].

[4] Adobe (2019). Buy Adobe Photoshop | Best photo, image, and design editing software. [online] Adobe.com. Available at: https://www.adobe.com/products/photoshop.html.

[5] I. J. Goodfellow et al., "Generative Adversarial Nets," Advances in Neural Information Processing Systems, pp. 2672–2680, 2014.

[6] Zhang, L., Li, C., Wong, T.-T., Ji, Y. and Liu, C. (2019). Two-stage sketch colorization. *ACM Transactions on Graphics*, 37(6), pp.1–14. doi:https://doi.org/10.1145/3272127.3275090.

[7] Zhang, R., Zhu, J.-Y., Isola, P., Geng, X., Lin, A.S., Yu, T. and Efros, A.A. (2017). Real-time user-guided image colorization with learned deep priors. *ACM Transactions on Graphics*, 36(4), pp.1–11. doi:https://doi.org/10.1145/3072959.3073703.

[8] Larsson, Gustav & Maire, Michael & Shakhnarovich, Gregory. (2016). Learning Representations for Automatic Colorization. 9908. 577-593. 10.1007/978-3-319-46493-0_35.

[9] Hensman, P. and Aizawa, K. (2017). cGAN-Based Manga Colorization Using a

Single Training Image. *arXiv (Cornell University)*. doi:https://doi.org/10.1109/icdar.2017.295.

[10] Petalica-paint.pixiv.dev. 2020. Petalica Paint. [online] Available at: <https://petalica-paint.pixiv.dev/index_en.html> [Accessed 2 Janurary 2024].

[11] Chie Furusawa, Kazuyuki Hiroshiba, Keisuke Ogaki and Odagiri, Y. (2017). Comicolorization. doi:https://doi.org/10.1145/3145749.3149430.

[12] Zhang, R., Isola, P. and Efros, A., 2016. Colorful Image Colorization. Computer Vision – ECCV 2016, pp.649-666.

[13] R. Ye, W. Wang, J. Chen and K. W. Lin, "Interactive Anime Sketch Colorization with Style Consistency via a Deep Residual Neural Network," 2019 International Conference on Technologies and Applications of Artificial Intelligence (TAAI), Kaohsiung, Taiwan, 2019, pp. 1-5,

[14] Zou, C., Mo, H., Gao, C., Du, R. and Fu, H., 2019. Language-based colorization of scene sketches. ACM Transactions on Graphics, 38(6), pp.1-16.

[15] Goodfellow, I.J., Pougetabadie, J., Mirza, M., Xu, B., Wardefarley, D., Ozair, S., Courville, A., Bengio, Y., Ghahramani, Z., Welling, M.: Generative adversarial nets. Advances in Neural Information Processing Systems 3, 2672–2680

[16] Putri, V. and Fanany, M., 2017. Sketch Plus Colorization Deep Convolutional Neural Networks for Photos Generation from Sketches. Proceeding of the Electrical Engineering Computer Science and Informatics, 4(1).

[17] L. Zhang, Y. Ji, X. Lin and C. Liu, "Style Transfer for Anime Sketches with Enhanced Residual U-net and Auxiliary Classifier GAN," 2017 4th IAPR Asian Conference on Pattern Recognition (ACPR), Nanjing, 2017, pp. 506-511.

[18] Kaggle.com. 2020. Anime Sketch Colorization Pair. [online] Available at: <https://www.kaggle.com/ktaebum/anime-sketch-colorization-pair> [Accessed 21 March 2024].

[19] Mirza, M. & Osindero, S. (2014), 'Conditional Generative Adversarial Nets' , cite arxiv:1411.1784

[20] Chen, W. & Hays, J. (2018), 'SketchyGAN: Towards Diverse and Realistic Sketch to Image Synthesis.', CoRR abs/1801.02753 .

[21] Ci, Y., Ma, X., Wang, Z., Li, H. and Luo, Z., 2018. User-Guided Deep Anime Line Art Colorization with Conditional Adversarial Networks. 2018 ACM Multimedia Conference on Multimedia Conference - MM '18,.

[22] Shi, Min & Zhang, Jia-Qi & Chen, Shuyu & Gao, Lin & Lai, Yu-Kun & Zhang, Fang-Lue. (2020). Deep Line Art Video Colorization with a Few References.

[23] Lee, Junsoo & Kim, Eungyeup & Lee, Yunsung & Kim, Dongjun & Chang, Jaehyuk & Choo, Jaegul. (2020). Reference-Based Sketch Image Colorization Using Augmented-Self Reference and Dense Semantic Correspondence. 5800-5809. 10.1109/CVPR42600.2020.00584.

[24] Yingge Qu, Tien-Tsin Wong, and Pheng-Ann Heng. 2006. Manga colorization. ACM Trans. Graph. 25, 3 (July 2006), 1214–1220. DOI:https://doi.org/10.1145/1141911.1142017

[25] S. Albawi, T. A. Mohammed and S. Al-Zawi, "Understanding of a convolutional neural network," 2017 International Conference on Engineering and Technology (ICET), Antalya, 2017, pp. 1-6, doi: 10.1109/ICEngTechnol.2017.8308186.

[26] Openaccess.thecvf.com. 2024. [online] Available at: <https://openaccess.thecvf.com/content_cvpr_2017/papers/Isola_Image-To-Image_Translation_Wi th_CVPR_2017_paper.pdf> [Accessed 7 February 2024].

[27] Keras (2019). *Home - Keras Documentation*. [online] Keras.io. Available at: https://keras.io.

[28] TensorFlow (2019). *TensorFlow*. [online] TensorFlow. Available at: https://www.tensorflow.org.

[29] Chintala, S. (2021). *soumith/ganhacks*. [online] GitHub. Available at: https://github.com/soumith/ganhacks.

[30] Radford, Alec, Luke Metz, and Soumith Chintala. "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks." *ArXiv*, (2015). Accessed April 27, 2024. /abs/1511.06434.

[31] Litjens, G., Kooi, T., Bejnordi, B.E., Setio, A.A.A., Ciompi, F., Ghafoorian, M., van der Laak, J.A.W.M., van Ginneken, B. and Sánchez, C.I. (2017). A Survey on Deep Learning in Medical Image Analysis. *Medical Image Analysis*, 42, pp.60–88. doi:https://doi.org/10.1016/j.media.2017.07.005.

[32] docs.opencv.org. (n.d.). *OpenCV: Histogram Comparison*. [online] Available at: https://docs.opencv.org/3.4/d8/dc8/tutorial_histogram_comparison.html.

[33] Dumoulin, Vincent, and Francesco Visin. "A Guide to Convolution Arithmetic for Deep Learning." *ArXiv*, (2016). Accessed April 27, 2024. /abs/1603.07285.

[35] Cao, Y., Meng, X., Mok, P.Y., Lee, T.-Y., Liu, X. and Li, P. (2024). AnimeDiffusion: Anime Diffusion Colorization. IEEE Transactions on Visualization and Computer Graphics, pp.1–14. doi:https://doi.org/10.1109/tvcg.2024.3357568.

[36] Liu, X., Zhu, S., Zeng, Y. and Zhang, J. (2023). Adversarial Interactive Cartoon Sketch Colourization with Texture Constraint and Auxiliary Auto‐Encoder. Computer Graphics Forum. doi:https://doi.org/10.1111/cgf.14797.

[37] Li, X. and Zhang, Y. (2022). Footwear Sketches Colorization Method based on Generative Adversarial Network. International journal of advanced computer science and applications/International journal of advanced computer science & applications, 13(12). doi:https://doi.org/10.14569/ijacsa.2022.0131219

[38]Liu, X. *et al.* (2021) 'Reference-guided structure-aware deep sketch colorization for cartoons', *Computational Visual Media*, 8(1), pp. 135–148. doi:10.1007/s41095-021-0228-6.

[39] Wang, N., Niu, M., Wang, Z., Hu, K., Liu, B., Wang, Z. and Li, H. (2023). Region Assisted Sketch Colorization. IEEE transactions on image processing, 32, pp.6142–6154. doi:https://doi.org/10.1109/tip.2023.3326682.

[40] Liu, X., Wu, W., Li, C., Li, Y. and Wu, H. (2021). Reference-guided structure-aware deep sketch colorization for cartoons. Computational Visual Media, 8(1), pp.135–148. doi:https://doi.org/10.1007/s41095-021-0228-6.

[41] Liu, G., Chen, X. and Hu, Y., 2019. Anime Sketch Coloring with Swish-gated Residual U-net and Spectrally Normalized GAN. Engineering Letters, 27(3).

APPENDIX A - CODE LISTING

1. Convolutional Block

```
def convolutional_block(input, filters, fsize=4, stride=2, dropout=False, activation=True, BatchNormalization=True, padding='same'):
    convolutional_layer = Conv2D(filters, fsize, stride, padding, kernel_initializer=tf.keras.initializers.truncated_normal(stddev=.02))(input)
    if BatchNormalization:
        convolutional_layer = BatchNormalization()(convolutional_layer)
    if activation:
        convolutional_layer = LeakyReLU(alpha=0.2)(convolutional_layer)
    return convolutional_layer
```

2. Deconvolutional Block

```
def deconvolutional_block(input, filters, fsize=4, stride=2, out=None, dropout=False, activation=True, BatchNormalization=True, padding='same'):
    deconvolutional_layer = Conv2DTranspose(filters, fsize, stride, padding, kernel_initializer=tf.keras.initializers.truncated_normal(stddev=.02))(concatenate([input, out]) if out is not None else input)
if BatchNormalization:
        deconvolutional_layer = BatchNormalization()(deconvolutional_layer)
    if dropout:
        deconvolutional_layer = Dropout(rate=dropout)(deconvolutional_layer)
```