2022 IEEE International Symposium on Circuits and Systems (ISCAS) | 978-1-6654-8485-5/22/$31.00 ©2022 IEEE | DOI: 10.1109/ISCAS48785.2022.9937519

Anime Character Recognition using Intermediate Features Aggregation

Edwin Arkel Rios*, Min-Chun Hu†, Bo-Cheng Lai*
*National Yang Ming Chiao Tung University, Hsinchu, Taiwan
†National Tsing Hua University, Hsinchu, Taiwan

Abstract—**In this work we study the problem of anime character recognition. Anime, refers to animation produced within Japan and work derived or inspired from it. We propose a novel Intermediate Features Aggregation classification head, which helps smooth the optimization landscape of Vision Transformers (ViTs) by adding skip connections between intermediate layers and the classification head, thereby improving relative classification accuracy by up to 28%. The proposed model, named as *Animesion*, is the first end-to-end framework for large-scale anime character recognition. We conduct extensive experiments using a variety of classification models, including CNNs and self-attention based ViTs. We also adapt its multimodal variation Vision-Language Transformer (ViLT), to incorporate external tag data for classification, without additional multimodal pre-training. Through our results we obtain new insights into the effects of how hyperparameters such as input sequence length, mini-batch size, and variations on the architecture, affect the transfer learning performance of Vi(L)Ts.**

1. Introduction

Anime, originally a word to describe animation works produced in Japan, can be seen now as an umbrella term for work that is inspired or follows a similar style to the former [1]. It is a complex, global, cultural phenomenon, with an industry that surpasses 2 trillion Japanese yen [2]. Recently, the anime film, Kimetsu no Yaiba (Demon Slayer), became the highest-grossing film of all time in Japan and the 5th highest-grossing film of 2020 worldwide [3]. Clearly anime as a phenomenon and industry is thriving from an economic point of view. Furthermore, viewing has been recognized as an integral part of literacy development by educators [4], and its importance as a medium cannot be understated. For these reasons, it is imperative for content streaming platforms such as Netflix to develop robust multimedia content analysis systems for more efficient access, digestion, and retrieval of information.

We leverage ViTs for the task of anime character recognition. ViTs, like CNNs are sensitive to mini-batch size selection [5], [6], [7], [8], but to a much higher degree. However, they differ from CNNs in that intermediate feature maps are uniform across all layers. Taking advantage of this, we propose a simple but effective Intermediate Features Aggregation (IFA) classification head which helps improve the performance of ViTs across a variety of experimental settings. Second, we perform a study on the effects of a variety of design hyperparameters (base model architecture, input sequence size, mini-batch size) on the recognition accuracy of anime characters. Lastly, we release our source-code and pretrained model checkpoints, in an effort to encourage and facilitate researchers to continue work in this domain. This paper is organized as follows: Section 2 presents relevant work on computer vision and computational methods for drawn media while Section 3 describes the experimental methodology. Then, in Section 4 we discuss the results and in Section 5 we summarize our findings.

2. Background and Related Work

2.1. Transformers for Computer Vision

Transformers [11] have become the state-of-the-art (SotA) in Natural Language Processing (NLP) and therefore in the past few years there's been quite active research into porting this architecture for vision tasks [12], [13]. The big breakthrough came in the form of the Vision Transformer (ViT) [9]. In their paper, Dosovitskiy et al. took a transformer encoder and applied it directly to image patches, beating the current SotA, in both classification accuracy and computation efficiency, in a variety of benchmarks. However, transformers suffer from quadratic costs in memory requirements with respect to the input size, and when it comes to images, this quickly makes for models which require industrial-level hardware to fine-tune for a given task, let alone pretrain from scratch.

For this reason, many works have studied how to improve vision transformers efficiency by making significant changes to the architecture or the attention mechanism [14], [15]. However, there's been less studies on how to improve the performance and efficiency from the training strategy and hyperparameters side. Touvron et al. [16] studied the influence of the optimizer, data augmentation and regularization, and noticed how transformers are sensitive to the settings of optimization hyperparameters. This influence of hyperparameters for neural network training has been widely studied for computer vision using CNNs [17], and also for transformers in NLP settings [18]. Among these hyperparameters, asides from the optimization algorithm and the learning rate, there's the (mini-)batch size. In general, smaller mini-batch sizes may lead to better performance, in terms of classification accuracy, due to consistent arrival at

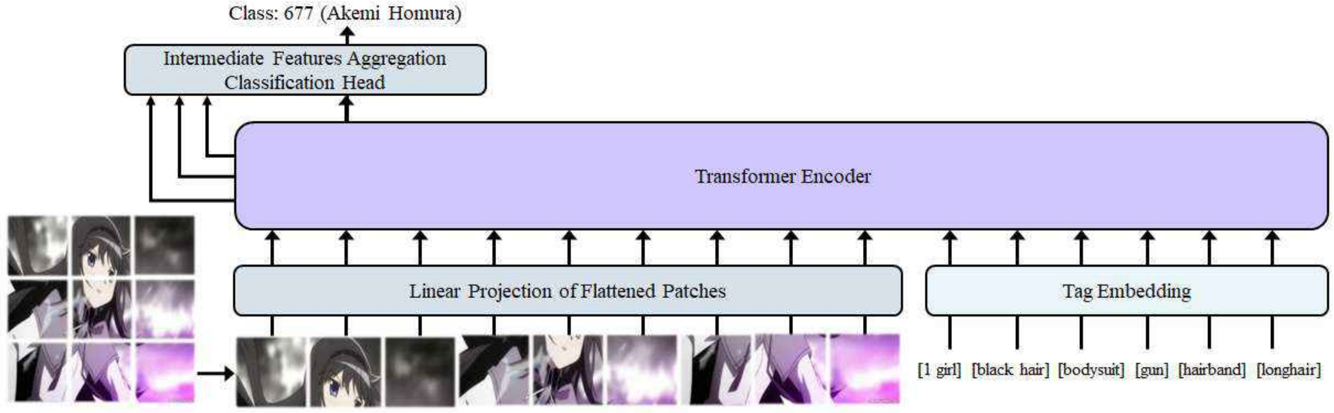

Class: 677 (Akemi Homura)
Intermediate Features Aggregation Classification Head
Transformer Encoder
Linear Projection of Flattened Patches
Tag Embedding
[1 girl]
[black hair]
[bodysuit]
[gun]
[hairband]
[longhair]

Figure 1. Overview of our framework for anime character recognition. Figure adapted from ViT [9] and ViLT [10]

flat minima, in contrast to the sharp minima at which large batch sizes training leads to [5], [6], [7].

However, in practice, the consensus is that we aim to maximize performance while minimizing training time. Due to this, many methods have been proposed to leverage parallelization, by increasing batch size, while keeping competitive performance. Goyal et al. [19] suggested to use a learning rate that is a function of the batch size, along with learning rate warm-up. Yet, another important consideration is the accessibility to hardware. While some, in practice have access to unlimited computation nodes, others are much more constrained in terms of computational resources. If we aim to fully utilize the capabilities of vision transformers for a variety of tasks, we need to optimize these hyperparameters to obtain a better trade-off between performance, and cost, both in terms of computational and monetary resources.

2.2. Computer Vision for Drawn Media

Anime, comics, cartoons, manga, and sketches, all of these have something in common; traditionally, they have all been drawn media. Drawn media has significant differences compared to natural images, images taken by common RGB cameras, which most CV algorithms are designed for. In particular, most drawn media is not as texture rich as photos, and this may affect CNNs performance since they may be biased towards textures, rather than shapes [20]. For this reason, drawn media can be a challenging testbed for CV models. Therefore, we aim to evaluate ViT models in this task that may be challenging for CNNs.

CV research on these mediums is not new and several reviews on approaches leveraging computation exist [21]. Most of the existing works have been focused on how to apply CV methods for image translation, synthesis, generation and/or colorization of characters [22], [23]. However, the task of character recognition and classification has been mostly unexplored. We aim to gain further understanding in this area of research on drawn media as we believe that hierarchical learning, from simple to complex, with a better understanding of what makes a certain character unique, will allow us to design better automatic character generative models down the line, as has been demonstrated through class-guided generation [24], [25], and semantically consistent translation with natural images [26].

3. Methodology

3.1. Data

We use the *DanbooruAnimeFaces* dataset in our experiments. *DAF* [27], is a subset of the 2018 release of *Danbooru20xx* [28]. Due to its extremely long-tailed distribution, we only keep classes with at least 20 samples, resulting in 463, 437 images of 3,263 characters. We split it into training, validation, and testing sets using a ratio of 0.7, 0.1, and 0.2, respectively. Since the original dataset only contains face crops, we also sample full body images by resizing the original images from *Danbooru20xx*, and coin it as *DAFull*. Furthermore, we include description tags from *Danbooru20xx* as additional multimodal data.

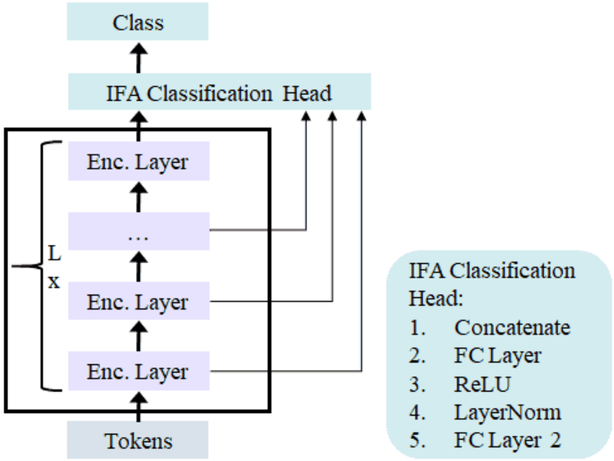


Figure 2. Diagram of proposed Intermediate Features Aggregation (IFA) classification head.

3.2. Experiments

Architecture. We conduct experiments on *DAF* and *DAFull*, using image sizes of 128x128 and mini-batch size of 64, using both CNN-based architectures (ResNet-18 and ResNet-50 [29], and EfficientNet-B0 [30]) and vision transformers (ViT B-16, B-32, L-16, L-32), with and without pretraining. All of the ViT models were pretrained on *ImageNet-21k* (*IN-21k*), following the procedure described in [9], while CNN-based ones are pretrained on *ImageNet-1k*.

Mini-batch and image size. We study the effect of image size and mini-batch size when fine-tuning vision transformers. We test five different batch sizes: 256, 128, 64, 32, and 16, and two image sizes: 128x128, and 64x64.

Multimodal inputs. We study the possibility of using tag descriptions as text tokens to enhance classification performance. For these experiments we utilize the modifications proposed by Kim et al. in ViLT [10] to process text data using a ViT. We tokenize the text strings utilizing a pretrained BERT [31] WordPiece (WP) tokenizer from the HuggingFace [32] library. Utilizing this tokenizer leads to an average of 39.5 tokens for each image, and for this reason, we utilize 32 text tokens as input as baseline for our multimodal ViLT experiments. We compare the multimodal versions, against the vision-only, for both *DAF* and *DAFull*. We also study the effect of the number of text tokens used as input, by comparing against 16 and 64 text tokens.

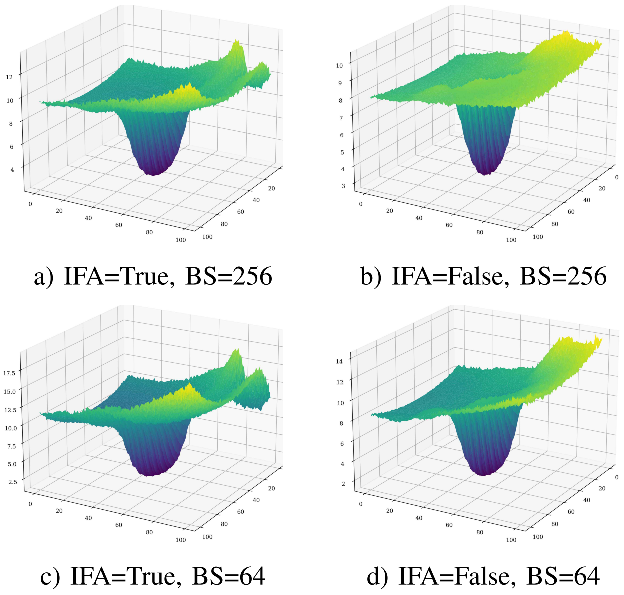
IFA classification head. We propose a simple modification to the classification head used in the Vision Transformer, by extracting the best features from all intermediate layers, as shown in Figure 2. Instead of just using the [CLS] token (the first token from ViT) from the last layer as input to the classification head, we concatenate the [CLS] tokens from all layers ($L = 12$ in the case of ViT B models) into a third dimension ($B \times D \times L$, where B is the batch size and D is the hidden dimension size of the transformer) and pass the concatenated features through a fully connected (FC) layer that reduces the number of dimensions back to two ($B \times D$), followed by a ReLU activation function, and a LayerNorm, then pass it through a second FC layer that changes the dimensions of the output into the number of classes used for classification. We visualize the effects of IFA head by utilizing *loss-landscapes* package [33] inspired by [8].

For all of our experiments we utilize stochastic gradient descent (SGD) with momentum, with an initial learning rate (LR) of 0.001 and momentum of 0.9, and train for 50 epochs. For the experiments on architecture, mini-batch and image size, we also apply a constant epoch (CE) LR decay, where we reduce the current LR by 1/3 after each 20 epochs. For the experiments using multimodal inputs, and using the proposed intermediate feature aggregation classification head, we utilize a cosine LR scheduler with warm-up of 1000 steps. For data augmentation, we apply random crop, where we first resize the image to a square of size 160x160 or 96x96, then take a random square crop of the desired input size (128, and 64 respectively). Additionally, we perform random horizontal flip, color jittering, and normalization of the images. We normalize using mean and standard deviation of 0.5 for all three RGB channels. For validation and testing, we only resize the images to the desired input size, and normalize them. If not specified, we utilize image size 128x128 and mini-batch size 64. We compare results utilizing test set top-1 average classification accuracy (%) and when needed, the relative percentage change compared to a baseline.

4. Results and Discussion

TABLE 1. CLASSIFICATION ACCURACY FOR *DAF* AND *DAFull* USING DIFFERENT ARCHITECTURES, WITH(OUT) PRETRAINING.

Model	Pretrained=False		Pretrained=True	
	DAF	DAFull	DAF	DAFull
RN-18	*75.00*	*67.68*	81.16	74.68
RN-50	**76.41**	**69.64**	*84.33*	**80.37**
EffN-B0	72.75	64.47	80.37	74.21
ViT B-16	58.99	34.93	83.78	76.10
ViT B-32	47.30	29.45	75.49	61.55
ViT L-16	56.52	35.95	**87.70**	*79.38*
ViT L-32	49.65	30.39	76.31	62.58



a) IFA=True, BS=256 b) IFA=False, BS=256

c) IFA=True, BS=64 d) IFA=False, BS=64

Figure 3. Visualization of loss surfaces for ViT B-32.

We highlight the best results in bold, and in certain cases, the second best, in italics.

Character recognition. CNN-based architectures perform much better in the absence of pretraining, but as more data and computational resources are allocated, transformers tend to outperform CNNs (Table 1).

The type and size of the input has a large impact on classification performance when using transformers. There's a big performance gap when utilizing *DAF* vs *DAFull*, this is due the fact that by inputting the face area we are directly facilitating the model a discriminative region. A module for automatic extraction of discriminative regions would help alleviate this issue, and make the model more practical. The image size is another important factor since it's directly

TABLE 2. CLASSIFICATION ACCURACY FOR *DAF* AS A FUNCTION OF MINI-BATCH SIZE FOR IMAGE SIZE 128X128 AND 64X64.

Model	Batch size	16		32		64		128		256	
	Image size	128	64	128	64	128	64	128	64	128	64
ViT B-16		**87.71**	72.24	86.23	69.63	83.78	62.67	78.21	50.80	64.52	33.41
ViT B-32		**80.40**	49.83	79.62	47.09	75.49	39.62	66.78	28.82	49.64	18.93
ViT L-16		**89.71**	73.69	88.78	73.11	87.70	70.52	85.34	62.92	78.36	47.04
ViT L-32		**81.64**	50.42	80.13	47.09	76.31	40.14	68.48	28.92	52.36	18.14

TABLE 3. EFFECTS OF INPUTTING TAGS AS ADDITIONAL INFORMATION, FOR *DAF* AND *DAFull*.

Model	Dataset			
	DAF		*DAFull*	
	Top-1	Difference	Top-1	Difference
ViLT B-16	85.93	+2.57	81.26	+6.78
ViLT B-32	81.67	**+8.19**	74.34	**+20.78**
ViLT L-16	**90.30**	+2.96	**85.44**	+7.63
ViLT L-32	81.03	+6.19	73.31	+ 17.15

TABLE 4. EFFECTS OF TEXT SEQUENCE LENGTH IN CLASSIFICATION ACCURACY, FOR *DAF* AND *DAFull*.

Set	Model	Max text tokens		
		16	32	64
DAF	ViLT B-16	85.68	85.93	**86.14**
	ViLT B-32	79.74	81.67	**82.3**
DAFull	ViLT B-16	79.74	81.26	**81.61**
	ViLT B-32	71.04	74.34	**75.61**

correlated to the effective input sequence length, and it specially affects the [B/L]-32 models, as their effective input sequence length is more than halved due to the larger patch size, as can be seen in Table 2

Similarly, adding other modalities of inputs such as tags helps boost the performance (Table 3), as it increases the effective input sequence length, and enriches it with attributes that represent additional data for the model to extract useful correlations. Likewise, increasing the max text input length also increases the classification accuracy (Table 4).

Surprisingly, the mini-batch size may be the strongest predictor for performance of a given transformer model, with all other hyperparameters fixed, smaller batch sizes consistently outperform larger ones (Table 2).

TABLE 5. EFFECTS OF PROPOSED IFA CLASSIFICATION HEAD IN CLASSIFICATION ACCURACY, FOR *DAF* AND *DAFull*.

Model	Dataset			
	DAF		*DAFull*	
	Top-1	Difference	Top-1	Difference
ViT B-16	86.22	+2.91	78.44	+3.07
ViT B-32	77.70	+2.93	63.41	+3.02
ViLT B-16	**89.17**	**+3.77**	**84.37**	+3.83
ViLT B-32	84.40	+3.34	78.05	+ **4.99**

TABLE 6. ACCURACY AND PERFORMANCE CHANGE FOR *DAF* WHEN USING IFA CLASSIFICATION HEAD, AS FUNCTION OF MINI-BATCH SIZE.

Model	Mini-batch size				
	16	32	64	128	256
B-16	**87.57**	87.07	86.22	83.06	74.19
	0	+1.36	+3.92	+8.39	**+19.41**
B-32	**80.15**	79.56	77.70	72.85	63.09
	-0.62	+0.52	+3.61	+10.00	**+28.39**

Finally, in Table 5 we can see how our simple IFA classification head leads to improvement across tested datasets, architectures (and effective input sequence length), and modalities, while incurring an almost negligible computational overhead of 1 extra minute per epoch (19 min vs 18 min), and roughly 1.3 GB extra VRAM usage (14.4 GB vs 13.1 GB), when using ViT B-16 with batch size 256. From Table 6 we can see how our proposed IFA helps reduce the gap between different batch sizes, and in Figure 3 we can see a visualization of the loss surface around the local minima after training. We can see how larger batch sizes make the minima sharper but our IFA head helps smooth them out.

5. Conclusion

In this work we study a variety of factors that affect anime character recognition performance, including pretraining strategy, architecture variations, input sequence size and type, mini-batch size, and classification head. Our proposed IFA classification head effectively helps reduce the sensitivity of ViTs to mini-batch size, recovering lost performance as batch size is increased, while incurring minimal computation overhead. Our results also demonstrate the feasibility of using multimodality augmented transformers in a relatively low-resource setting, without extensive additional pretraining.

6. Acknowledgements

This work was supported by the Ministry of Science and Technology, Taiwan, under grant MOST 108-2628-E-009-007-MY3. We thank NCHC for providing computational and storage resources.

References

[1] P. Brophy, *Tezuka the Marvel of Manga*. Melbourne, Vic: National Gallery of Victoria, Jan. 2007.

[2] "Anime Industry Data | ." [Online]. Available: https://aja.gr.jp

[3] D. Harding, "Demon Slayer: Mugen Train Dethrones Spirited Away to Become the No. 1 Film in Japan of All Time." [Online]. Available: https://www.crunchyroll.com/anime-news/2020/12/27-1/demon-slayer-mugen-train-dethrones-spirited-away-to-become-the-no-1-film-in-japan-of-all-time

[4] N. Frey and D. Fisher, *Teaching Visual Literacy: Using Comic Books, Graphic Novels, Anime, Cartoons, and More to Develop Comprehension and Thinking Skills*. Corwin Press, Jan. 2008, google-Books-ID: cb4xcSFkFtsC.

[5] M. Hardt, B. Recht, and Y. Singer, "Train faster, generalize better: Stability of stochastic gradient descent," *arXiv:1509.01240 [cs, math, stat]*, Feb. 2016, arXiv: 1509.01240. [Online]. Available: http://arxiv.org/abs/1509.01240

[6] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, "On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima," *arXiv:1609.04836 [cs, math]*, Feb. 2017, arXiv: 1609.04836. [Online]. Available: http://arxiv.org/abs/1609.04836

[7] D. Masters and C. Luschi, "Revisiting Small Batch Training for Deep Neural Networks," *arXiv:1804.07612 [cs, stat]*, Apr. 2018, arXiv: 1804.07612. [Online]. Available: http://arxiv.org/abs/1804.07612

[8] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein, "Visualizing the Loss Landscape of Neural Nets," *arXiv:1712.09913 [cs, stat]*, Nov. 2018, arXiv: 1712.09913. [Online]. Available: http://arxiv.org/abs/1712.09913

[9] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," *arXiv:2010.11929 [cs]*, Oct. 2020, arXiv: 2010.11929. [Online]. Available: http://arxiv.org/abs/2010.11929

[10] W. Kim, B. Son, and I. Kim, "ViLT: Vision-and-Language Transformer Without Convolution or Region Supervision," *arXiv:2102.03334 [cs, stat]*, Feb. 2021, arXiv: 2102.03334. [Online]. Available: http://arxiv.org/abs/2102.03334

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," *arXiv:1706.03762 [cs]*, Dec. 2017, arXiv: 1706.03762. [Online]. Available: http://arxiv.org/abs/1706.03762

[12] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in Vision: A Survey," *arXiv:2101.01169 [cs]*, Jan. 2021, arXiv: 2101.01169. [Online]. Available: http://arxiv.org/abs/2101.01169

[13] N. Parmar, A. Vaswani, J. Uszkoreit, Kaiser, N. Shazeer, A. Ku, and D. Tran, "Image Transformer," *arXiv:1802.05751 [cs]*, Jun. 2018, arXiv: 1802.05751. [Online]. Available: http://arxiv.org/abs/1802.05751

[14] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows," *arXiv:2103.14030 [cs]*, Mar. 2021, arXiv: 2103.14030. [Online]. Available: http://arxiv.org/abs/2103.14030

[15] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction without Convolutions," *arXiv:2102.12122 [cs]*, Aug. 2021, arXiv: 2102.12122. [Online]. Available: http://arxiv.org/abs/2102.12122

[16] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," *arXiv:2012.12877 [cs]*, Jan. 2021, arXiv: 2012.12877. [Online]. Available: http://arxiv.org/abs/2012.12877

[17] L. N. Smith, "A disciplined approach to neural network hyper-parameters: Part 1 – learning rate, batch size, momentum, and weight decay," *arXiv:1803.09820 [cs, stat]*, Apr. 2018, arXiv: 1803.09820. [Online]. Available: http://arxiv.org/abs/1803.09820

[18] M. Popel and O. Bojar, "Training Tips for the Transformer Model," *The Prague Bulletin of Mathematical Linguistics*, vol. 110, no. 1, pp. 43–70, Apr. 2018, arXiv: 1804.00247. [Online]. Available: http://arxiv.org/abs/1804.00247

[19] P. Goyal, P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He, "Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour," *arXiv:1706.02677 [cs]*, Apr. 2018, arXiv: 1706.02677. [Online]. Available: http://arxiv.org/abs/1706.02677

[20] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel, "ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness," *arXiv:1811.12231 [cs, q-bio, stat]*, Jan. 2019, arXiv: 1811.12231. [Online]. Available: http://arxiv.org/abs/1811.12231

[21] O. Augereau, M. Iwata, and K. Kise, "A survey of comics research in computer science," *arXiv:1804.05490 [cs]*, Apr. 2018, arXiv: 1804.05490. [Online]. Available: http://arxiv.org/abs/1804.05490

[22] Y. Jin, J. Zhang, M. Li, Y. Tian, H. Zhu, and Z. Fang, "Towards the Automatic Anime Characters Creation with Generative Adversarial Networks," *arXiv:1708.05509 [cs]*, Aug. 2017, arXiv: 1708.05509. [Online]. Available: http://arxiv.org/abs/1708.05509

[23] L. Zhang, Y. Ji, and X. Lin, "Style Transfer for Anime Sketches with Enhanced Residual U-net and Auxiliary Classifier GAN," *arXiv:1706.03319 [cs]*, Jun. 2017, arXiv: 1706.03319. [Online]. Available: http://arxiv.org/abs/1706.03319

[24] P. Dhariwal and A. Nichol, "Diffusion Models Beat GANs on Image Synthesis," *arXiv:2105.05233 [cs, stat]*, Jun. 2021, arXiv: 2105.05233. [Online]. Available: http://arxiv.org/abs/2105.05233

[25] S. Gopalakrishnan, P. R. Singh, Y. Yazici, C.-S. Foo, V. Chandrasekhar, and A. Ambikapathi, "Classification Representations Can be Reused for Downstream Generations," *arXiv:2004.07543 [cs, stat]*, Apr. 2020, arXiv: 2004.07543. [Online]. Available: http://arxiv.org/abs/2004.07543

[26] A. Cherian and A. Sullivan, "Sem-GAN: Semantically-Consistent Image-to-Image Translation," *arXiv:1807.04409 [cs]*, Jul. 2018, arXiv: 1807.04409. [Online]. Available: http://arxiv.org/abs/1807.04409

[27] Y. Wang, "grapeot/Danbooru2018AnimeCharacterRecognitionDataset," Dec. 2020, original-date: 2019-07-01T19:19:32Z. [Online]. Available: https://github.com/grapeot/Danbooru2018AnimeCharacterRecognitionDataset

[28] G. Branwen, "Danbooru2019: A Large-Scale Crowdsourced and Tagged Anime Illustration Dataset," Dec. 2015, last Modified: 2020-09-04. [Online]. Available: https://www.gwern.net/Danbooru2019

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *arXiv:1512.03385 [cs]*, Dec. 2015, arXiv: 1512.03385. [Online]. Available: http://arxiv.org/abs/1512.03385

[30] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," *arXiv:1905.11946 [cs, stat]*, Sep. 2020, arXiv: 1905.11946. [Online]. Available: http://arxiv.org/abs/1905.11946

[31] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *arXiv:1810.04805 [cs]*, May 2019, arXiv: 1810.04805. [Online]. Available: http://arxiv.org/abs/1810.04805

[32] "huggingface/transformers," Aug. 2021, original-date: 2018-10-29T13:56:00Z. [Online]. Available: https://github.com/huggingface/transformers

[33] M. D. Bernardi, "loss-landscapes," Sep. 2021, original-date: 2019-03-16T15:03:18Z. [Online]. Available: https://github.com/marcellodebernardi/loss-landscapes