AniVid: A Novel Anime Video Dataset with Applications in Animation

Kai Gangi
Stuyvesant High School
kgangi20@stuy.edu

Abstract

Automating steps of the animation production process using AI-based tools would ease the workload of Japanese animators. Although there have been recent advances in the automatic animation of still images, the majority of these models have been trained on human data and thus are tailored to images of humans. In this work, I propose a semi-automatic and scalable assembling pipeline to create a large-scale dataset containing clips of anime characters' faces. Using this assembling strategy, I create AniVid, a novel anime video dataset consisting of 34,221 video clips. I then use a transfer learning approach to train a first order motion model (FOMM) on a portion of AniVid, which effectively animates still images of anime characters. Extensive experiments and quantitative results show that FOMM trained on AniVid outperforms other trained versions of FOMM when evaluated on my test set of anime videos.

1. Introduction

The Japanese anime industry has been experiencing steady growth for the past seven years, gaining significant traction around the world. The global anime market size was valued at 20.47 billion USD in 2018 and is expected to reach 36.26 billion USD by 2025 [8]. However, animators in Japan are plagued by long work hours and low wages, which discourages many from pursuing this career. According to a 2018 study conducted by the Japan Animation Creators Association, Japanese animators worked an average of 230 hours per month, while the overall Japanese population worked an average 132 hours per month [11].

To ease the workload of Japanese animators and accelerate the animation production process, animators can utilize AI-based tools for assistance [19]. For example, recent advances have been made in the task of automatically animating a still image by extracting the movements from a driving video. The first order motion model (FOMM) has achieved state-of-the-art results for this task without the need for annotated data [22]. In this paper, I propose the implementation of this model on anime characters in order to convert images of these characters to fully-animated shots.

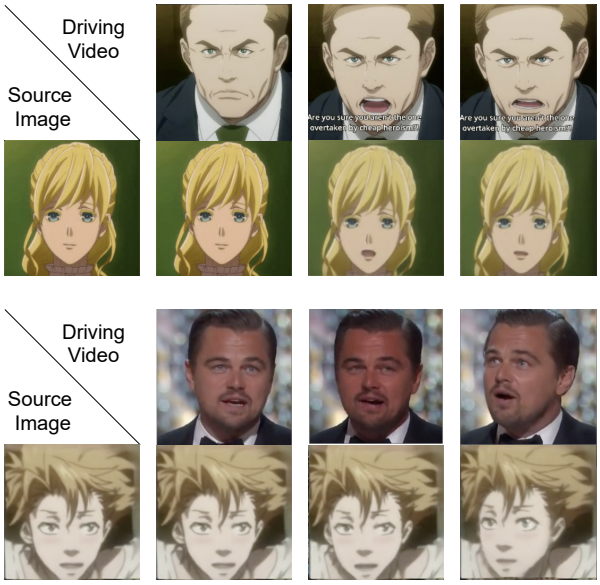

Figure 1. Example animations from FOMM after continuing the training process on `AniVid`. I test this model using anime-based driving videos (top) and human-based driving videos (bottom).

The success of FOMM requires the availability of significant domain-specific data for training. To the best of my knowledge, large-scale public video datasets of anime characters currently do not exist. Therefore, I create a novel dataset for this task titled `AniVid` using a semi-automatic assembling pipeline (see Figure 2). FOMM is then trained on `AniVid`, producing example animations such as those shown in Figure 1. Extensive experiments and quantitative results (see Table 2) show that training this model on `AniVid` outperforms other trained versions of FOMM when evaluated on my test set of anime videos.

The contributions of my work are three-fold, and are summarized below. The dataset and trained models will be publicly released in the near future.

- I propose a semi-automatic and scalable assembling pipeline that can construct a video dataset containing characters' faces. Using this pipeline, I create

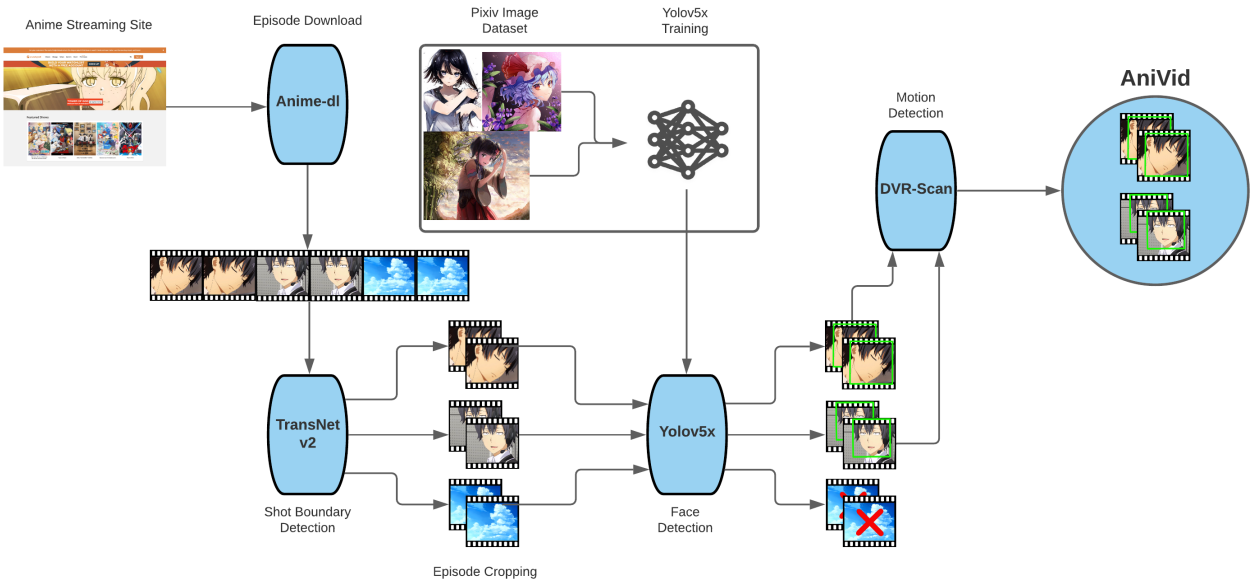

Figure 2. The semi-automatic pipeline I use to construct AniVid.

AniVid, a novel and diverse anime video dataset compiled from 19 different anime series.

- In the process of constructing AniVid, I train a YOLOv5x object detection model on a dataset of anime face images.
- I train FOMM on AniVid in order to automatically animate still images of anime characters.

2. Related Work

Many AI-based methods have been proposed to accelerate the anime production process. GAN-based approaches have been heavily utilized in this realm, such as in the tasks of sketch-editing [23, 17]. SGA-GAN [16] and PSGAN [10] have been employed to generate completely original anime character faces, and a model based on DRAGAN [12] has been able to generate full anime character bodies. For the task of line-art colorization, Seg2Pix [21] and GANs with generators based on U-net [7, 29] have achieved state of the art results.

Although the results of these works have been robust, they all focus on automating processes done on individual drawings. With anime episodes containing an average of 3000 drawings [27], there is still a huge workload to be had even with certain processes being automatic. However, using my implementation of FOMM, one singular image can be animated into multiple seconds of content, vastly reducing the number of total drawings needed to create an episode.

3. Dataset Collection Pipeline

This section describes the semi-automatic pipeline for automatically collecting and processing a large-scale video dataset of anime faces, starting from full-length anime episodes downloaded from online providers. The videos come in MPEG encodings and H.264 containers, with an average framerate of 23 fps and average length of 3.74 seconds. Refer to Figure 2 for a visual diagram of the proposed pipeline.

3.1. Preliminary Data Download

AniVid requires videos of talking anime heads, so dialogue-heavy programs are preferred over action-heavy ones. Thus, I primarily choose series within the romance and comedy genre. Utilizing the anime-dl [1] command line program, I download a collection of 19 different anime series.

3.2. Shot Boundary Detection

A shot is defined as a series of connected frames captured by single camera in a continuous time frame [9]. Furthermore, detecting the frame locations between two consecutive shots is known as shot boundary detection. We implement an off-the shelf TransNetv2 model for this task [24]. Each full-length episode is then split into shorter video clips based on the location of the detected shot boundaries, which effectively increases the size of the dataset.

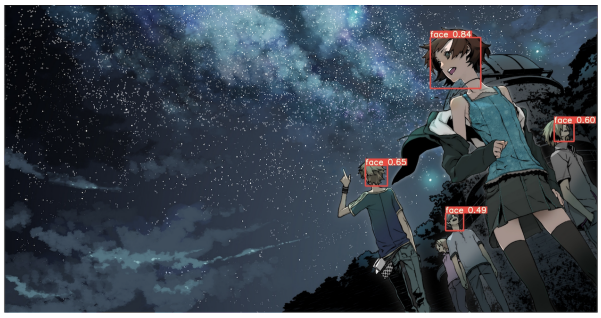

Figure 3. One of my test set images after implementing face detection on it. YOLOv5x is able to detect multiple faces at far distances and at odd angles.

3.3. Face Detection

To ensure that all the videos in my dataset have faces within them, I require the use of a face detection model. Therefore, I train and evaluate a YOLOv5x object detection model [13] on an annotated image dataset of anime characters (see Figure 3) [28]. Once the training process is complete, I implement this model on each video clip, deleting any frames without a detected face. I also remove any frames containing multiple faces to ensure uniformity in the dataset.

Loss Function Face detection loss is calculated from bounding box regression score, objectness score, and class probability score. For the bounding box regression score, I utilize Generalized Intersection over Union (GIoU) loss [20]. Unlike Intersection over Union (IoU), GIoU can account for the relative distance between predicted and ground truth bounding boxes even when their intersection is zero. Assuming predicted and ground truth bounding boxes A and B, the GIoU is computed using the equation below:

$$\begin{aligned} GIoU &= IoU - \frac{|C \backslash (A \cup B)|}{|C|} \\ &= \frac{|A \cap B|}{|A \cup B|} - \frac{|C \backslash (A \cup B)|}{|C|} \end{aligned} \tag{1}$$

where C is the smallest convex hull that encloses A and B [20]. For calculating the objectness score and class probability score, I use Binary Cross-Entropy with Logits Loss Function, a built-in function within the PyTorch library [2].

3.4. Motion Detection

I use the DVR-Scan [3] command line application to detect motion in each video clip with a threshold score of 0.35. Shots with a motion score below this threshold were removed from the dataset. FOMM generates motion animations, so videos without motion would not be suitable for training.

4. FOMM Task and Procedure

FOMM animates a singular image based on the motions from a driving video without requiring any annotations or prior information on the object being animated [22]. The original developers of the model have trained it separately on the following datasets: `Bair`, `Fashion-Videos`, `Tai-Chi-HD`, `AniVid`, `Nemo`, and `VoxCeleb` [4].

I propose a transfer learning approach, continuing the training process on the `VoxCeleb` weights. I choose `VoxCeleb` over the other datasets due to its higher performance on my `AniVid` test set (see Table 2). Additionally, `VoxCeleb` most closely resembles the content I desire for the videos in `AniVid`, as it contains videos of talking human faces. It is also important to note that anime characters have many visual traits similar to real humans, otherwise known as anthropomorphism. Therefore, by starting the training process using these pretrained human weights, a great deal of time is saved compared to training the model from scratch with randomly initialized weights. Further investigating this correlation between virtual media and realistic human images remains as a topic for future work.

5. Experiments

5.1. Face Detection

Experimental Setup I use the YOLOv5x model pretrained on MS COCO [13] and continue the training process on a dataset of 6,173 images of anime characters compiled from Pixiv [28]. Data augmentation methods (scaling, color space adjustments, mosaic augmentation) are performed on the training data. The base learning rate is 0.01 and SGD is used as the optimizer. Upon training the model, I filter out any frames without faces from each video chunk, using a conservative confidence threshold of 0.60 to minimize the number of false positives. Google Colab Pro was used to run the code.

Evaluation Protocol To evaluate the performance of this face detection model, I use four metrics: mAP@0.5, mAP@[.5:.05:.95], precision, and recall. According to MS COCO's definition of mAP for object detection [6, 14], if there is an interpolated precision-recall curve $p(r)$, the AP@0.5 is defined as the mean of the precision values on a set of 101 equally spaced recall values (0 to 1 at step size of .001), where an IoU of 0.5 or greater is considered a true positive. Furthermore, the mAP@0.5 is simply the mean of the AP@0.5 of each class. This is summarized in the equation below, where j is some IoU threshold and C is the total number of classes:

$$mAP@j = \frac{1}{101C} \sum_{i=1}^{C} \sum_{r \in \{0, 0.01, \ldots 1\}} \max_{\tilde{r}: \tilde{r} \geq r} p(r) \tag{2}$$

Metric	mAP@0.5	mAP@[.5:.05:.95]	Precision	Recall
Value (%)	97.96	71.77	98.27	95.96

Table 1. Results after evaluating YOLOv5x on my test set of Pixiv illustrations.

Metrics	Bair	Fashion	UvA-NEMO	Taichi	VoxCeleb	AniVid
SAM	44.50	22.67	43.03	43.37	78.07	**80.92**
PSNR	22.00	20.19	22.78	22.95	42.40	**45.47**
SRE	25.56	44.32	25.93	26.02	47.33	**50.14**
FSIM	0.2168	0.1927	0.2253	0.2352	0.4733	**0.5481**
SSIM	0.4913	0.4739	0.5029	0.5030	0.9156	**0.9607**
UIQ	0.0941	0.06409	0.0895	0.1182	0.2602	**0.3543**

Table 2. Results after evaluating different trained versions of FOMM on my test set of anime videos (the higher the number, the better).

mAP@[.5:.05:.95] refers to the mean of the mAPs at IoU thresholds ranging from 0.5 to 0.95 with a step size of 0.05. The equation for this is shown below, with a being the minimum IoU threshold, b being the maximum IoU threshold, and γ being the step-size.

$$mAP@[a:\gamma:b] = \sum_{j=a/\gamma}^{b/\gamma} j\gamma \cdot mAP@j \tag{3}$$

Results Table 1 presents the results of evaluating YOLOv5x on my test set of Pixiv anime images, and an example detection can be seen in Figure 3. YOLOv5x has achieved greater performance than other existing anime face detectors. Faster-RCNN had a mAP of 0.9086 when evaluated on a very similar set of Pixiv images [28], and Fast-RCNN, Faster-RCNN, and SSD respectively had mAPs of 0.810, 0.816, and 0.765 when evaluated on the `Manga109` dataset [18].

5.2. FOMM

Experimental Setup I first select 200 random videos from `AniVid`. These videos were manually cropped into 256×256 squares, with the character's face at the center. Using the pretrained weights from the `VoxCeleb` dataset, I continue the training process on these 200 videos for an additional 100 epochs. Google Colab Pro was used to run the code. Furthermore, training on the full `AniVid` dataset would require more extensive hardware, making it a task for future work.

Evaluation Protocol To evaluate the performance of FOMM, I first perform the task of video reconstruction on my test set. This is done by implementing FOMM on the first frame of each video, using the video itself as the driving video. I then compare each frame from the ground-truth video to the generated video using six image similarity metrics: spectral angle mapper (SAM) [30], peak signal-to-noise ratio (PSNR) [5], signal to reconstruction error ratio (SRE) [15], feature-based similarity index (FSIM) [31], structural similarity index (SSIM) [26], and universal image quality index (UIQ) [25].

Results Table 2 summarizes the results from implementing the original releases of FOMM that were pretrained on non-anime datasets and the FOMM that I continued training on `AniVid`. Continuing the training process on `AniVid` improved all metrics. Example animations after training on `AniVid` are shown in Figure 1.

6. Conclusion

I propose a dataset creation pipeline that is used to generate a novel anime video dataset titled `AniVid`. By continuing the training process of FOMM on a portion of `AniVid`, I am able to animate still images of anime faces. In hindsight, if I use live footage of humans as driving videos, this model could also have applications for Virtual YouTubers. In the future, I plan to train the model on the full `AniVid` dataset, further investigate the effects of integrating human data with anime data using a domain adaptation framework, and integrate live video footage into FOMM.

References

[1] `https://github.com/anime-dl/anime-downloader`.

[2] `https://pytorch.org/docs/stable/generated/torch.nn.BCEWithLogitsLoss.html`.

[3] `https://dvr-scan.readthedocs.io/en/latest/`.

[4] Siarohin Aliaksandr. First order motion model for image animation. `https://github.com/AliaksandrSiarohin/first-order-model`, 2019.

[5] Kalpana Chauhan, Rajeev K. Chauhan, and Anju Saini. Chapter 5 - enhancement and despeckling of echocardiographic images. In Nilanjan Dey, Amira S. Ashour, Fuqian Shi, and Valentina E. Balas, editors, *Soft Computing Based*

Medical Image Analysis, pages 61–79. Academic Press, 2018.

[6] COCO Common Objects in Context.

[7] Tzu-Ting Fang, Duc Minh Vo, Akihiro Sugimoto, and Shang-Hong Lai. Stylized-colorization for line arts. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 2033–2040, 2021.

[8] Grand View Research, Inc. Anime market size, share trends analysis report by type (t.v., movie, video, internet distribution, merchandising, music, pachinko, live entertainment), by region, and segment forecasts, 2019 - 2025. Technical Report GVR-3-68038-841-1, CA, USA, 2019. [Online].

[9] D. S. Guru, Mahamad Suhil, and P. Lolika. A novel approach for shot boundary detection in videos. *CoRR*, abs/1608.06716, 2016.

[10] Koichi Hamada, Kentaro Tachibana, Tianqi Li, Hiroto Honda, and Yusuke Uchida. Full-body high-resolution anime generation with progressive structure-conditional generative adversarial networks. *CoRR*, abs/1809.01890, 2018.

[11] Japanese Animation Creators Association. Animation creator fact finding report 2018. Technical report, 2018. [Online].

[12] Yanghua Jin, Jiakai Zhang, Minjun Li, Yingtao Tian, Huachun Zhu, and Zhihao Fang. Towards the automatic anime characters creation with generative adversarial networks. *CoRR*, abs/1708.05509, 2017.

[13] Glenn Jocher, Alex Stoken, Jirka Borovec, NanoCode012, ChristopherSTAN, Liu Changyu, Laughing, tkianai, Adam Hogan, lorenzomammana, yxNONG, AlexWang1900, Laurentiu Diaconu, Marc, wanghaoyang0106, ml5ah, Doug, Francisco Ingham, Frederik, Guilhen, Hatovix, Jake Poznanski, Jiacong Fang, Lijun Yu , changyu98, Mingyu Wang, Naman Gupta, Osama Akhtar, PetrDvoracek, and Prashant Rai. ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements, Oct. 2020.

[14] Harshit Kumar. Evaluation metrics for object detection and segmentation: map. Technical Fridays, Sept. 20, 2019. [Online].

[15] Charis Lanaras, José M. Bioucas-Dias, Silvano Galliani, Emmanuel Baltsavias, and Konrad Schindler. Super-resolution of sentinel-2 images: Learning a globally applicable deep neural network. *CoRR*, abs/1803.04271, 2018.

[16] Hongyu Li and Tianqi Han. Towards Diverse Anime Face Generation: Active Label Completion and Style Feature Network. In *Eurographics 2019 - Short Papers*, 2019. [Online].

[17] Jia Li, Nan Gao, Tong Shen, Wei Zhang, Tao Mei, and Hui Ren. Sketchman: Learning to create professional sketches. MM '20, page 3237–3245, New York, NY, USA, 2020. Association for Computing Machinery.

[18] Yusuke Matsui, Kota Ito, Yuji Aramaki, Toshihiko Yamasaki, and Kiyoharu Aizawa. Sketch-based manga retrieval using manga109 dataset. *CoRR*, abs/1510.04389, 2015.

[19] Tatiana Mejia. State of ai in animation. Adobe Blog, Jun. 11, 2019. [Online].

[20] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union. June 2019.

[21] Chang Wook Seo and Yongduek Seo. Seg2pix: Few shot training line art colorization with segmented image data. *Applied Sciences*, 11(4).

[22] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. First order motion model for image animation. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[23] Edgar Simo-Serra, Satoshi Iizuka, and Hiroshi Ishikawa. Mastering sketching: Adversarial augmentation for structured prediction. *CoRR*, abs/1703.08966, 2017.

[24] Tomás Soucek and Jakub Lokoc. Transnet V2: an effective deep network architecture for fast shot transition detection. *CoRR*, abs/2008.04838, 2020.

[25] Zhou Wang and A.C. Bovik. A universal image quality index. *IEEE Signal Processing Letters*, 9(3):81–84, 2002.

[26] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[27] Washi. Anime production – detailed guide to how anime is made and the talent behind it! Washi's Blog, Jan. 18, 2011. [Online].

[28] Zhou Xuebin. Anime-face-detector. `https://github.com/qhgz2013/anime-face-detector`, 2018.

[29] Mingcheng Yuan and Edgar Simo-Serra. Line art colorization with concatenated spatial attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 3946–3950, 2021.

[30] R. H. Yuhas, A. Goetz, and J. Boardman. Discrimination among semi-arid landscape endmembers using the spectral angle mapper (sam) algorithm. 1992.

[31] Lin Zhang, Lei Zhang, Xuanqin Mou, and David Zhang. Fsim: A feature similarity index for image quality assessment. *IEEE Transactions on Image Processing*, 20(8):2378–2386, 2011.