Domain-adaptive Self-supervised Pre-training for Face & Body Detection in Drawings

by

Barış Batuhan Topal

A Dissertation Submitted to the
Graduate School of Sciences and Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in

Computer Science and Engineering

KOÇ ÜNİVERSİTESİ

July 30, 2022

Domain-adaptive Self-supervised Pre-training for Face & Body Detection in Drawings

Koç University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a master's thesis by

Barış Batuhan Topal

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

_____

Assoc. Prof. Metin Tevfik Sezgin (Advisor)

_____

Prof. Hazım Kemal Ekenel

_____

Prof. Engin Erzin

Date: _____

Dedicated to my family and everyone who cared and supported me even during hard times.

ABSTRACT

Domain-adaptive Self-supervised Pre-training for
Face & Body Detection in Drawings
Barış Batuhan Topal
Master of Science in Computer Science and Engineering
July 30, 2022

Drawings are powerful means of pictorial abstraction and communication. Understanding diverse forms of drawings, including digital arts, cartoons, and comics, has been a major problem of interest for the computer vision and computer graphics communities. Although there are large amounts of digitized drawings from comic books and cartoons, they contain vast stylistic variations, which necessitate expensive manual labeling for training domain-specific recognizers.

In this work, I show how self-supervised learning, based on a teacher-student network with a modified student network update design, can be used to build face and body detectors. My setup allows exploiting large amounts of unlabeled data from the target domain when labels are provided for only a small subset of it. I further demonstrate that style transfer can be incorporated into my learning pipeline to bootstrap detectors using a vast amount of out-of-domain labeled images from natural images (i.e., images from the real world). My combined architecture yields detectors with state-of-the-art (SOTA) and near-SOTA performance using minimal annotation effort.

Through the utilization of this detector architecture, I accomplish a set of additional tasks. First, I extract a large set of facial drawing images (∼1.2 million instances) from unlabeled data and train SOTA generative adversarial network (GAN) models to generate and a SOTA GAN inversion model to reconstruct faces. When the detector-aided data is leveraged, these generative models successfully learn diverse stylistic features. Secondly, I implement an annotation tool to enlarge the existing set of annotated data. This tool offers users to annotate bounding boxes of panels, speech bubbles, narrations, faces, and bodies; to associate text boxes with faces and bodies; to transcript the text; to match the same characters in the image.

ÖZETÇE

Yüksek Lisans Tez Başlığı
Barış Batuhan Topal
Bilgisayar Bilimi ve Mühendisliği, Yüksek Lisans
30 July 2022

Çizimler, resimsel soyutlama ve iletişim için güçlü araçlardır. Dijital sanatlar, karikatürler ve çizgi romanlar dahil olmak üzere bu çeşitli çizim biçimlerini anlamak, bilgisayarla görme ve bilgisayar grafiği toplulukları için büyük bir ilgi sorunu olmuştur. Dijital ortamda çok sayıda çizim özellikle çizgi roman ve çizgi film şeklinde erişilebilir olsa da, bu örnekler geniş stilistik farklılıklar içermekte olduğundan, alana özel tanıma modellerinin eğitiminde pahalı etiketleme süreçleri gerekmektedir.

Bu çalışmada, değiştirilmiş öğrenci ağı güncelleme dizaynına sahip bir öğretmen öğrenci ağına dayalı öz denetimli öğrenmenin yüz ve vücut dedektörleri oluşturmak için nasıl kullanılabileceğini gösterdim. Kurulumum, yalnızca az sayıda etiketlenmiş çizim resimleri seti kullanarak çok miktardaki etiketlenmemiş veriden faydalanmayı sağlıyor. Ayrıca, stil aktarma yöntemlerini öğrenme sürecine ekleyerek alan dışı doğal hayat resimlerinden faydalanmanın detektör performansını daha da arttırdığını kanıtladım. Kombine mimari, minimum etiketleme çabası kullanarak son teknoloji çizim detektörlerinden daha iyi ya da kıyaslanabilir performansa erişim sağlamakta.

Bu detektör yapısından faydalanarak birtakım ekstra görevleri de tamamladım. İlk olarak, etiketlenmemiş veriler üzerinden geniş bir yüz çizim resim seti çıkarttım ($\sim$1.2 milyon örnek) ve yüz yaratmak için son teknoloji üretken çekişmeli ağ (GAN) modellerini, yeniden inşa etmek için ise son teknoloji bir GAN inversiyon modeli eğittim. Detektör destekli veriden yararlanıldığı takdirde, üretken modeller başarılı bir şekilde çeşitli stilistik özellikleri öğreniyor. İkinci olarak, var olan etiketlenmiş veriyi genişletmek için bir etiketleme aracı geliştirdim. Bu araç kullanıcılara panel, konuşma balonu, anlatım, yüz ve vücut sınırlayıcı kutularını etiketleme; yazıları yüz ve vücutlarla eşleştirme; yazıları transkripte dökme; resimdeki aynı karakterleri eşleştirme imkanı sunuyor.

ACKNOWLEDGMENTS

I would like to express my gratitude to Assoc. Prof. T. Metin Sezgin and Prof. Deniz Yuret for their guidance. Their support led me to a project on drawings, which I have delightedly worked on in the past two years. Their command in this area helped me to overcome various problems and steer in the right direction during my study.

Although I graduated in Computer Science and Engineering during my Bachelor's, I did not know much about Deep Learning. Without the assistance of my advisors, professors, and the courses I have taken, this thesis would have never existed. I am immensely grateful for being a student at Koç University. I also like to thank KUIS AI for its financial support, Prof. Hazım Kemal Ekenel and Prof. Engin Erzin for being on my thesis committee.

I would like to thank all my friends in our Intelligent User Interfaces (IUI) Lab for their companionship. Our discussions have always motivated me, broadened my horizon, and made these two years much more fun. Special thanks to Gürkan Soykan for being the best teammate ever. I am grateful for all the projects we completed together and for encouraging each other even in the most challenging times.

My deepest gratitude goes to my family for always believing in me. I could not be here without their support and unconditional love.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

SOTA	state-of-the-art
AP	Average Precision
FID	Fretchet Inception Distance
KID	Kernel Inception Distance
LPIPS	Learned Perceptual Image Patch Similarity
MSE	Mean Squared Error
NMS	Non-maximum Suppression
FP & FN	False Positive & False Negative
SS	Self-supervised
ST	Style transferred
TN & SN	Teacher Network & Student Network
UI	User Interface
SGD	Stochastic Gradient Descend
EMA	Exponential Moving Average
iCF	iCartoonFace Dataset
M109	Manga 109 Dataset
DCM	DCM 772 Dataset
...-F	Faces Part of the Particular Dataset Given in "..."
...-B	Bodies Part of the Particular Dataset Given in "..."
C2k	Comic 2k Dataset
W2k	Watercolor 2k Dataset
C1k	Clipart 1k Dataset
C2k*	Mixture of C2k, W2k and C1k
eBD	eBDtheque Dataset

Chapter 1

INTRODUCTION

Drawings serve as a rich and expressive medium for communication. The earliest examples were painted on cave walls more than 45,000 years ago [Brumm et al., 2021]. Here I focus on comic books and cartoons, which are relatively recent forms of media. They combine text and graphics in a unique format to convey narratives. Key problems such as extracting the visual structure of the scenes, understanding the accompanying text, and modeling how they connect to form the narrative pose significant challenges. Hence, understanding comics has been a problem of interest to the computer vision, computer graphics, and NLP communities.

In drawings, the story is narrated primarily through the scene's main characters. Hence, in this thesis study, I focus on face and body detection, two primary problems for understanding drawings. Training face and body detectors is complicated by two challenges. First, although a tremendous amount of unlabeled data is available (primarily as digitized comic book pages and animations), face and body annotations are largely lacking. Second, since character design and drawing style change substantially across artists, series, and cultures, each domain inevitably requires domain-specific tuning to create detectors. In this thesis study, I present a pre-training pipeline for creating domain-adapted detectors, which addresses both problems. My pipeline has two major components. The first is a self-learning component that can exploit vast amounts of unlabeled data from the target domain to create detectors that can be tuned with minimal labeled data. I show that this self-learning model works best if it starts with a sufficiently good teacher. This component leads to the second key component of my pipeline, which uses style transfer to transform vast amounts of labeled natural images to create sufficiently

good teacher models.

There is existing work on teacher-student-based self-supervised learning methods, where the student, the teacher, and the final detector all operate in the same domain (natural images) [Xu et al., 2021, Liu et al., 2021]. Unlike these lines of work, I work across domains and show how images of natural scenes can be used to create detectors for drawings through style transfer and self-supervision. UMT [Deng et al., 2020] architecture also follows a similar path and works across domains, but my pipeline differs from this work in terms of the design of the self-supervision stage and the scope of style transfer. More specifically, I introduce a modified version of teacher-student architecture to drawings, where I periodically update student network's weights with teacher's after a specific number of iterations, and utilize the OHEM [Shrivastava et al., 2016] loss with an additional positive and negative confidence threshold limitation for a more stable training. Teacher and student networks are initialized using a model trained with cartoonized COCO [Lin et al., 2015], and WIDER FACE [Yang et al., 2016] transferred into 11 styles using methods from 4 style transfer algorithms [Wang and Yu, 2020, Hicsonmez et al., 2020, Zhu et al., 2017, Chen et al., 2018]. Even without drawing domain supervision, my teacher-student model outperforms previous supervised SOTA of DCM 772 [Nguyen et al., 2018] and weakly-supervised SOTA [Inoue et al., 2018] in most datasets.

I employ a multi-tasking strategy by jointly training the model for faces and bodies to reduce inference time and to benefit from the contextual and spatial relationship. To utilize datasets with face-only (e.g., [Zheng et al., 2020]) or body-only (e.g., [Inoue et al., 2018]) annotations, I use two detection heads: one to predict the faces, the other for bodies. When initialized with my pre-trained weights, my supervised model sets a new SOTA performance for most datasets, even if limited drawing data is used in training.

In addition to my pre-training pipeline, I leverage my final detector to construct a large-scale model-aided facial drawing images dataset that consists of $\sim$1.2 million faces from unlabeled drawings, train StyleGAN3 [Karras et al., 2021] and StyleGAN2 [Karras et al., 2020] with this newly generated dataset for face generation, and train HFGI [Bai et al., 2022] for face reconstruction tasks. Since the amount of training

images is large, the models successfully learn diverse stylistic features. Furthermore, I also design an annotation tool to enlarge the existing labeled data in this domain that enables annotating bounding boxes of panels, speech bubbles, narrations, faces, and bodies; associating text boxes with faces and bodies; transcribing the text; matching the same characters in the image.

Chapter 2 deals with the *Related Works* (i.e., face & object detectors, and style transferring studies, and the datasets). Chapter 3 explains my pre-training with style transferring & teacher-student network, fully-supervised fine-tuning, and additional tasks (i.e., annotation tool and face generation). In Chapter 4, my results and ablation studies are discussed.

Chapter 2

RELATED WORK

2.1 Face and Body Detection

Object detection models are the precursors of both face and body detection architectures. Typical object detection architectures include single-stage (e.g., YOLO [Redmon et al., 2015], SSD [Liu et al., 2016]) and two-stage detectors (e.g., Faster-RCNN [Ren et al., 2016]).

Models specific to face detection obtain the best results using a single-stage architecture with more model depth (e.g., TinaFace [Zhu et al., 2021]), likely because of the simpler nature of face detection compared to general object detection.

Body detection has primarily been studied as a sub-task in object detection models. With the increasing size of annotated data, models with high dependence on supervision were able to get good results [Bochkovskiy et al., 2020, Zhang et al., 2020b, Ge et al., 2021]. Unbiased Teacher [Liu et al., 2021] and Soft Teacher [Xu et al., 2021] introduced teacher-student training schemes and gained significant performance boost with a low amount of labeled data. Unlike this study, these studies target natural images. Thus, cross-domain detection with these models is prone to false positives (FP) and negatives (FN). Several studies have improved the teacher-student scheme to work well in cross-domain detection. While MTOR [Cai et al., 2019] exploits object relations in region-level consistency, inter-graph consistency and intra-graph consistency, UMT [Deng et al., 2020] tries to eliminate teacher and student network biases through distillation and style transferring, D-adapt [Jiang et al., 2021] adopts an adversarial pipeline to the detector model. Although my solution is more similar to UMT compared to other cross-domain studies, I improve its style transferring part by mixing multiple styles, I modify the standard teacher-student training to compensate for the FP and FN cases, and I change the loss

function to force the model to learn from more confident proposals.

Several studies have been done on face and object detection, specifically in drawings. Zhang *et al.* [Zhang et al., 2020a] proposed a fully-supervised face detector using only iCartoonFace, Ogawa *et al.* [Ogawa et al., 2018] trained a detector from Manga109, Nguyen *et al.* [Nguyen et al., 2018] used DCM772, Inoue *et al.* [Inoue et al., 2018] utilized Comic2k, Watercolor2k, and Clipart1k. In this study, I show that the performance on drawings can be significantly improved by using an effective pre-training pipeline and a better detector architecture.

2.2 Style Transfer

I use style transfer techniques to obtain pre-training data from natural images. Conversion of natural images to drawings is an unpaired image-to-image translation task. SOTA models for this task have been designed with Generative Adversarial Networks with U-Net-like generators (i.e., down-sampling first and then up-sampling). In my study, I use several cartoonization models (i.e., CycleGAN [Zhu et al., 2017], CartoonGAN [Chen et al., 2018], GANILLA [Hicsonmez et al., 2020], and White-Box Cartoonization [Wang and Yu, 2020]) to increase the stylistic variety of my pre-training data and select 11 styles from these works: Monet, Van Gogh, Cezanne from CycleGAN; Shinkai, Hayao, Hosoda, Paprika from CartoonGAN; AS, KH, Miyazaki from GANILLA; and the default style in White-Box Cartoonization. While previous detection studies on drawings have also utilized style transfer methods [Inoue et al., 2018, Deng et al., 2020], I improve on these results by combining multiple styles and analyzing which styles increase the performance more. My style transfer results can be seen in Chapter 4.

2.3 Datasets

Digitization has made millions of unlabeled drawings reachable on the internet. Thousands of old comic book series (e.g., Golden Age Comics between the 1930s

- 1950s) have been published on several websites [1][2] and gathered as an unlabeled dataset named COMICS [Iyyer et al., 2017]. Newer series can be obtained through web crawling. Unfortunately, annotated datasets only comprise a small subset of this domain in terms of stylistic variety and quantity.

Regarding the stylistic distribution of labeled datasets, the majority of iCartoonFace [Zheng et al., 2020] is retrieved from Asian products (∼74%), Manga109 [Matsui et al., 2017] only covers Japanese Manga styles, DCM 772 [Nguyen et al., 2018] is limited to comics from Golden Age Era. Although Inoue *et al.* [Inoue et al., 2018] introduced three additional small datasets for bodies (i.e., Comic2k, Watercolor2k, and Clipart1k), the combination of these provides only 2,500 training images, and they also remain stylistically bound in their sub-domain (e.g., only watercolor images in Watercolor2k). Currently, none of the available datasets provide comprehensive stylistic coverage. In particular, contemporary US and Western comics have little if any annotated examples.

In terms of dataset quantity, iCartoonFace contains a significant amount of face data with its 50,000 training and 10,000 validation images taken from animations, comic books, and children's shows. The situation is a bit more challenging with body annotations: Manga109 has ∼21,000 page images, but the style is limited to black and white mangas. DCM 772 consists of only 772 images. Comic2k, Watercolor2k, and Clipart1k increase the total labeled data by 2,500 instances. Building a body detector for drawings that is not fragile to different styles is challenging using only these datasets. Hence, self-supervised approaches are essential for creating suitable models for target instances with unseen styles.

2.4 Face Generation and Reconstruction

Different studies are conducted to generate images such as generative adversarial architectures, variational and vector-quantized auto-encoders, flow-based, and diffusion models. However, GANs have achieved superior outcomes, especially in face

[1]Comic Book Plus: https://comicbookplus.com

[2]Digital Comic Museum: https://digitalcomicmuseum.com

generation tasks. Since the proposal of the first StyleGAN [Karras et al., 2018], SOTA models have adopted the latest version of this series of architectures to their models. Current SOTA on FFHQ (1024x1024) [Karras et al., 2018] (i.e., StyleGAN-XL [Sauer et al., 2022]) also uses StyleGAN3 [Karras et al., 2021] as their baseline.

Since GAN-based methods are only trained for generation, these models do not support reconstruction by default, unlike auto-encoders. However, several GAN-inversion models are studied to construct latent embeddings from original images, which can be used to generate original-like images from GANs. Among these, HFGI [Bai et al., 2022] leverages StyleGAN2 [Karras et al., 2020] and results in SOTA performance among inversion models.

In my thesis, I choose StyleGAN3 and StyleGAN2 to generate faces and HFGI to generate embeddings. I train these models with the *Comic Faces in the Wild* dataset that I created with my final detector architecture and report the quantitative and visual results.

Chapter 3

METHODOLOGY

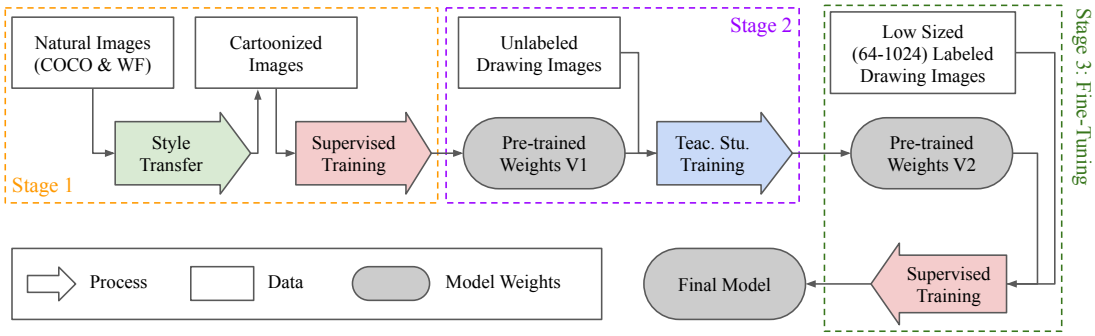

Figure 3.1: The complete pipeline

My model training consists of three stages: In the first stage, I use two large and annotated real-life image datasets, cartoonize them using style transfer methods, and perform pre-training for face and body detection. In the second stage, I utilize the extensive amount of unlabeled comic drawings available and perform the modified form of the teacher-student network training for self-supervision on my pre-trained model. In the final stage, I leverage the limited amount of annotated comic drawings to fine-tune my model. In Figure 3.1, you can see a demonstration of my complete pipeline. In the following subsections, I describe my base model, the three stages I propose, detector-aided *Comic Faces in the Wild* dataset, face generation task, and my annotation tool in more detail.

3.1 Model Architecture

While selecting my base model, I considered several aspects. Since the challenge in my data consists of stylistic variety in object representations (see Figure 3.2) , I

decided that adopting an object-detector-like model would provide greater performance, where the architecture is specifically designed to find multiple objects with various appearances. Secondly, I aimed to use a more robust and simple model with low inference time to focus mainly on the effects of style transfer and self-supervised training. Therefore, I decided to select one of the SOTA single-shot non-swin-transformer anchor-free object detectors, YOLOX [Ge et al., 2021], as my baseline architecture. However, my pre-training pipeline does not depend on this specific baseline. Hence it can be applied to any detector.

Figure 3.2: Examples on adversity of this domain (left: non-human character, middle: samples of different styles that characters are represented, right: a bird as a character in the upper side, birds as non-character items at the lower side)

As discussed in Chapter 2.3, some of the available drawing datasets do not include both face and body annotations together. Moreover, my style transferred datasets, namely COCO [Lin et al., 2014] and WIDER FACE [Yang et al., 2016], only include one or the other. To train my model jointly for both face and body parts and benefit from all the available datasets, I separate the detection head of the original YOLOX model into two pieces. Each piece proposes bounding boxes with their confidence values for only a single class. My overall architecture can be seen in Figure 3.3. During training, the heads are trained alternately at each forward pass.

In recent object detectors, Focal Loss [Lin et al., 2018] has been commonly used with good success. The default YOLOX model also uses a modified version of this loss, where it improves the label assignment technique by proposing the SimOTA

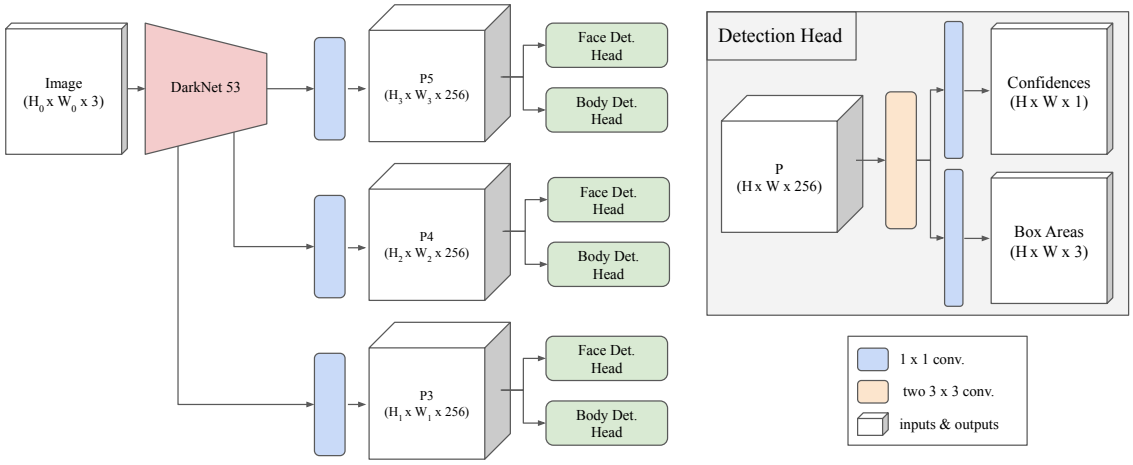

Figure 3.3: Complete model architecture

[Ge et al., 2021] method. I also adopt this structure in pre-training stage 1 and fine-tuning for my supervised training except for the class-specific losses since each of my heads only predicts a single class. However, in the self-supervised stage, I utilize a different loss function, which is explained in detail in Chapter 3.3.

3.2 Stage 1: Pre-training with Style Transferred Images

3.2.1 Preprocessing

I process the complete COCO [Lin et al., 2014] and WIDER FACE [Yang et al., 2016] dataset with 11 different styles as explained in Section 2.2. I eliminate all the images in COCO that do not have people or animals. I also count animals as bodies during training because drawings may include animal-like characters. To the best of my knowledge, no dataset includes annotations for animal faces. Thus, facial training is solely done through human faces in WIDER FACE. I discard the images in which a person has a face with its maximum facial side length smaller than $\sim$2% of the image's minimum side length. These faces are not required in my dataset since characters in drawings mostly have a bigger appearance on the image.

3.2.2 Training Experiments

I create 5 different experiments to test my model's success at pre-training stage 1:

- **Single Styles:** I analyze the effects of each of the 11 styles on the detection performance by training individual models with only one style transferring method.

- **All Styles:** I train an additional model by combining all styles with random selection per each image to notice if using multiple styles increases the overall performance.

- **Best Styles:** I choose five styles that result in the greatest performance individually and train another model by combining only these to find if selecting the most effective styles is more logical instead of utilizing all styles.

- **No Style:** I train an extra model that uses the original COCO and WIDER FACE images without any stylization to observe the benefit of style transferring.

- **No Animals Included:** Lastly, I test the effect of including animal bodies to body annotations to the performance. While all the experiments above uses animal annotations, for this specific experiment I utilize all of the styles but exclude the animal boxes from the training data.

3.3 Stage 2: Self-supervised Training with Teacher-student Network

3.3.1 Model Architecture

The model consists of two different network parts: teacher and student. These networks are identical and initialized from the same pre-trained set of weights that I obtain from the first style transferred pre-training stage by utilizing the mixture of all styles. The teacher network processes a non-augmented complete image and generates bounding box predictions with their confidence values. The student network also generates predictions, but it processes a heavily augmented version of the

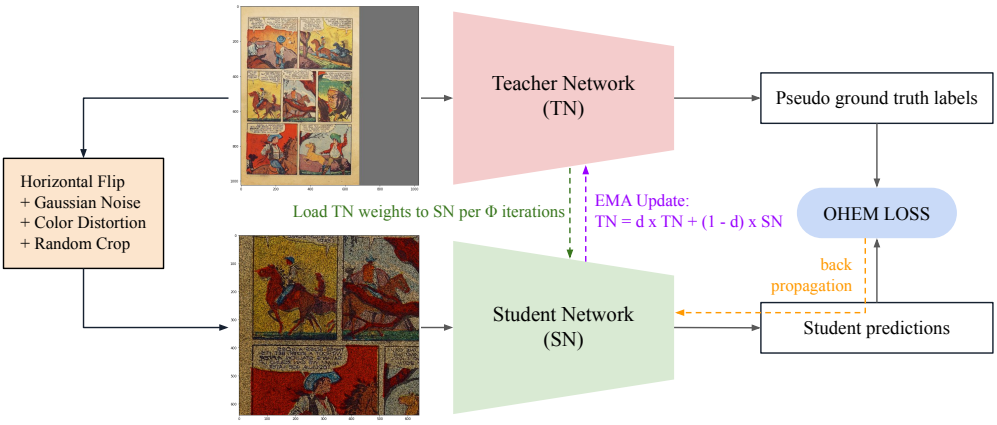

Figure 3.4: Teacher-student network training process

same input image. High-confidence predictions of the teacher network are further processed with the non-maximum suppression (NMS) algorithm, and the outputs are considered as the ground truth labels of the image. The student network is trained with the loss computed by using the artificial ground truth labels retrieved from the teacher network. The gradient flow of the teacher network is stopped, and it is updated at each iteration with respect to Eq. 3.1, where TN is the teacher, SN is the student network weight, d is a hyper-parameter.

$$TN = d \cdot TN + (1 - d) \cdot SN \tag{3.1}$$

Although TN is updated with the student weights in earlier studies, student weights are only changed with backpropagation. In my experiments, I have seen that this design causes the development of both modules at the earlier stages but a significant performance drop in SN in the later iterations due to the noisy pseudo ground truth labels caused by the change in the input domain between pre-training stage 1 (i.e., cartoonized natural images) and self-supervised processes (i.e., drawings). This drop also affects the performance of TN. Hence, I load the weights of TN to SN per each Φ iteration to fix the deterioration of SN. Since this step manipulates the values without the gradient flow, an optimizer with the momentum information

may mislead the overall model. Thus, I change my optimizer to Stochastic Gradient Descent (SGD).

3.3.2 Loss

In Focal Loss, each prediction is included in the confidence loss calculation with a weight that balances the positive (i.e., predictions in which the actual ground truth object is present) and negative (i.e., predictions that point to a background area) boxes. This approach is advantageous in fully supervised training since the ground truth box areas of every object in the image are given to the model. On the other hand, in self-supervised detectors, the high-probability predictions of the teacher model are selected as pseudo-ground truth values, which are prone to false positives (FP) and false negatives (FN). The existing works execute supervised and self-supervised stages in the same natural image set. This reduces the error rate of the teacher model. In my case, I pre-train the teacher model with cartoonized natural images and execute the self-supervised stage with drawings. Hence, the model is more vulnerable to errors. FP cases can be minimized by increasing the confidence threshold for ground truth selection. However, this choice also increases the FN rate. To further decrease the FN cases, I follow the OHEM loss [Shrivastava et al., 2016], where only a subset of predictions are chosen to calculate the loss. I also modify this loss so that the predictions can be selected as positive predictions only above a specific confidence threshold and negative predictions below a particular threshold. Subset selection and this modification help the model to skip a subset of FN cases of the teacher model in loss calculations (e.g., if a face/body area is predicted but has a low confidence value). Loss calculation of a single selected box proposal can be seen in Eq. 3.4:

$$\mathcal{L}_{conf} = -p \cdot ct_{pos} \cdot log(\hat{p}) - (1-p) \cdot ct_{neg} \cdot log(1-\hat{p}) \tag{3.2}$$

$$\mathcal{L}_{reg} = \sum_{i}^{\{w,h,x,y\}} smooth_{L_1}(i_{gt}, i_{pred}) \tag{3.3}$$

$$\mathcal{L}_{total} = \mathcal{L}_{conf} + \beta \mathcal{L}_{reg} \tag{3.4}$$

$\mathcal{L}_{conf}$ is the confidence loss and $\mathcal{L}_{reg}$ is the regression loss. $p \in \{0, 1\}$ indicates if the box is selected as positive ($p = 1$) or negative ($p = 0$), $\hat{p} \in [0, 1]$ is the confidence value of the selected box, $ct_{pos} \in \{0, 1\}$ is 1 if the confidence of the proposed box is above the positive confidence threshold, $ct_{neg} \in \{0, 1\}$ is 1 if the confidence of the proposed box is below the negative confidence threshold, $\{w, h, x, y\}$ are the width, height, and the center points of the box, β is the balancing parameter between confidence and regression losses.

3.3.3 Unlabeled Datasets

I crawl 195,321 comic book pages from today's US and European series to train my model. I also utilize 198,657 pages from COMICS and leveraged iCartoonFace, Manga109 pages, Comic2k, Watercolor2k, and Clipart1k images. At each forward pass, I select a random image from these image sets with the individual dataset probabilities that are set considering their sizes and stylistic coverage (see Table 3.1).

Table 3.1: Probability of selecting an image from specific datasets to the batch in stage 2

	Crawled	**COMICS**	**Manga109**	**iCartoonFace**	**Comic2k**	**Watercolor2k**	**Clipart1k**
Prob.	0.6	0.15	0.12	0.11	0.008	0.008	0.004

3.3.4 Experiments & Hyper-parameters

I run several experiments with different losses, Φ, β, d, positive and negative student confidence thresholds. The scores retrieved from these experiments can be seen in Table 4.2. As my final model, I set Φ to 500, β to 2, d to 0.9996, and positive and negative thresholds to 0.5.

3.4 Stage 3: Fully-supervised Training

I follow the model architecture described in Section 3.1 and conduct experiments with three different pre-training methods: random initialization, style transferred pre-training in stage 1, and teacher-student network from stage 2. Since each drawing dataset contains its own separate stylistic distribution, they should be fine-tuned separately to obtain the maximum performance on their test set. Thus, I fine-tune the model with single datasets for each pre-training variation by randomly selecting a limited number of image instances (i.e., 64, 128, 256, 512, 1024 images, or all data). As Manga109 and DCM772 consist of page images instead of individual panels, I separate panels during training to increase the number of input data and test the models with their page images.

3.5 Comic Faces in the Wild Dataset

Table 3.2: Number of face image instances from the individual datasets included in the Comic Faces in the Wild dataset

Resolutions	**iCartoonFace**	**Manga109**	**COMICS**	**Crawled**	**Total**
64 x 64	15,701	44,840	412,304	201,684	674,529
128 x 128	12,940	20,332	123,564	279,577	436,413
256 x 256	5,148	3,575	9,816	97,915	116,454
512 x 512	53	11	263	14,123	14,450
1024 x 1024	5	0	4	445	454
All x All	33,847	68,758	545,951	593,744	1,242,300
%	2.72	5.53	43.95	47.80	100.00

As mentioned in the Subsection 2.3 and 3.3.3, there exist multiple drawing data sources. To generate this model-aided dataset, I select the largest four: crawled images, COMICS, iCartoonFace, and Manga109. Using the best-performing version

of the detector, I extract square crops of all faces with a maximum side length greater than 32, a confidence score higher than 0.85, and an NMS value lower than 0.2. Afterward, I scale these crops to the closest resolution among 64 x 64, 128 x 128, 256 x 256, 512 x 512, and 1024 x 1024. Table 3.2 shows the number of faces belonging to each dimension.

3.6 Face Generation & Reconstruction

After the generation of the *Comic Faces in the Wild* dataset, I train both StyleGAN2 and StyleGAN3 for 128 x 128 and 256 x 256 resolutions. These resolutions are achieved by resizing all of the faces in the dataset to the corresponding dimensions. Although the quality of the images in 64 x 64 resolution is degraded after this step, I believe that using them with images of higher resolution will still be enough to preserve the overall generated image quality. Since my dataset is also facial, I use the default hyper-parameters provided for FFHQ [Karras et al., 2018]. Because of the GPU-related constraints, the batch size is set as 16 in StyleGAN2 and StyleGAN3 as 8 in HFGI. HFGI is only trained for 256 x 256 resolution and utilizes the final StyleGAN2 weights.

3.7 Annotation Tool

The tool is designed to be a simple interface for users to annotate:

- Bounding boxes of comic panels, faces, bodies, speech bubbles, and narratives
- The same characters in the given image
- Speech bubble-face-body association of the same character
- Transcription of the text boxes

Default Python3 libraries are preferred to implement the background processing, and Tkinter[1] is utilized to design the user interface (UI). Figure 3.5 is a sample image

[1]https://docs.python.org/3/library/tkinter.html

of the UI. The UI can also be accessed and used from this repository.

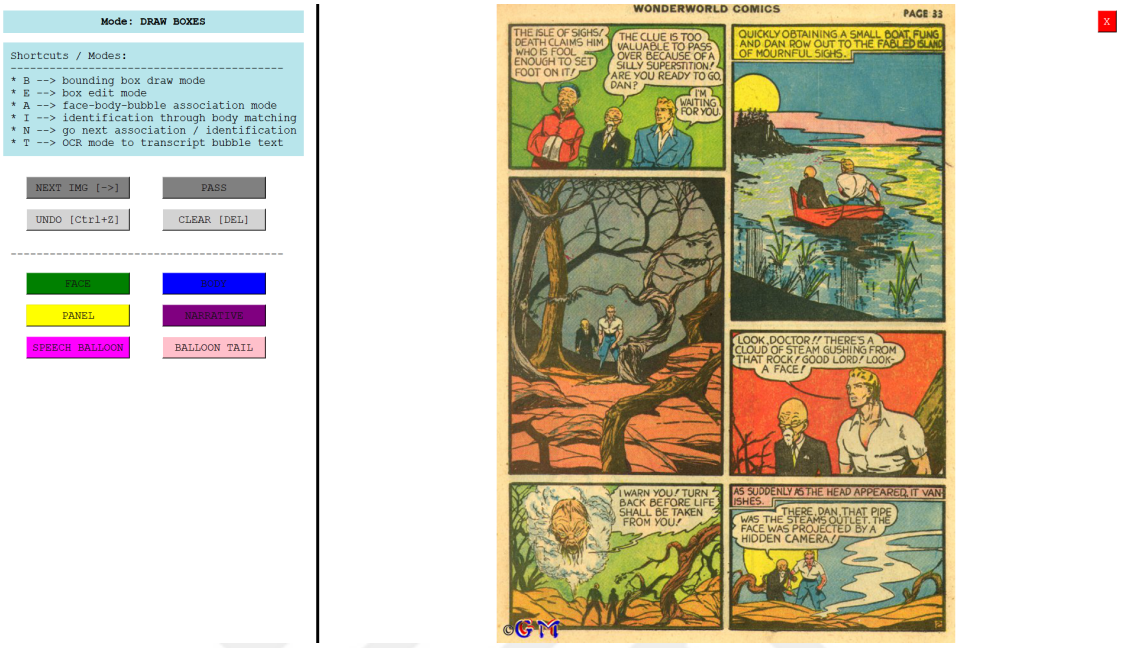

Figure 3.5: The complete interface of the annotation tool. No annotation has been done so far

The interface consists of six modes: box annotation, box editing, association, matching the same characters, going to the subsequent association/matching, and transcribing a text box. The transition between these modes is achieved by pressing the keyboard's corresponding keys. Following options with their shortcuts are also added:

- **"NEXT IMG":** When the labeling process of the current image is done, then the user can press it or use its shortcut (`RIGHT ARROW`) to proceed with the next image.

- **"CLEAR":** Clears the canvas. Shortcut is the `"DELETE"` key.

- **"UNDO":** Reverts the last operation done by the user. `"CTRL + Z"` is the shortcut.

- **"PASS":** If the user does not want to annotate the given image, this button passes the image without saving any annotation.

The complete control panel is placed on the left side of the interface. On the upper right there is a `"X"` button that can be used to exit the application.

3.7.1 Drawing Bounding Boxes

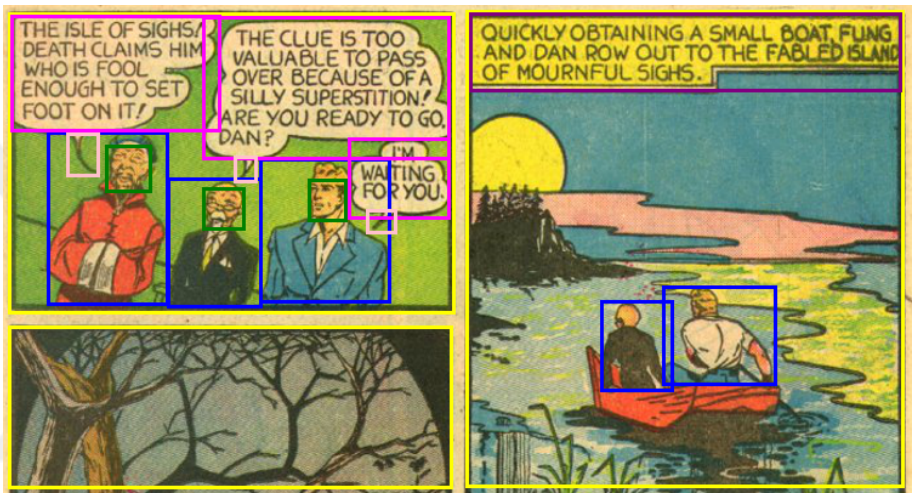

Figure 3.6: Annotations done in box mode

This mode is activated when the `"B"` key is pressed. Firstly, the user must click on the object button they want to annotate. These buttons are colored in the control panel on the left side. Figure 3.6 is an example after a set of bounding boxes is added to the image. To draw a bounding box, the user has to click on one corner of the box that will be added and then drag the mouse to the other corner. Starting from any corner is allowed.

3.7.2 Editing Bounding Boxes

Box editing mode is activated when the `"E"` key is pressed. With this mode, circles become visible on the corners of the boxes. By clicking on these circles, the user can change the width and height of the corresponding box. By clicking inside a box

and dragging, the user can also change the location of the drawn box. Figure 3.7 is an example of the UI where this mode is activated.

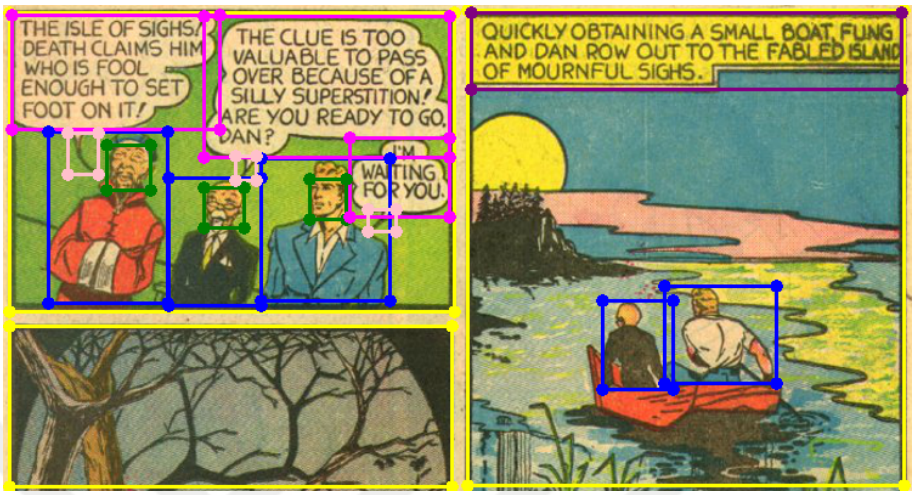

Figure 3.7: The UI when editing mode is activated

3.7.3 Adding Associations

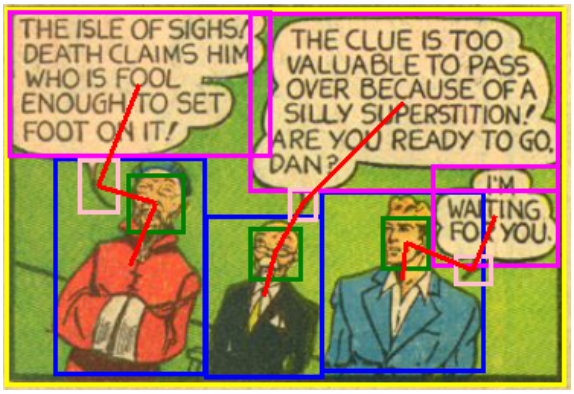

Figure 3.8: The UI when association mode is activated and the available associations are drawn

When the `"A"` key is pressed, association mode is activated. In this mode, user has to click on the boxes (i.e., speech bubble, speech bubble tail, face, and body boxes), where each object belongs to the same character. After finishing one

association, pressing `"N"` is mandatory to proceed with the next one. If there are 2 overlapping boxes in the area where the user clicks on, the smaller one is selected as the clicked box. The associations are drawn with red lines from the centers of one box to the other. Figure 3.8 demonstrates an example, where a list of associations are drawn.

3.7.4 Matching Characters

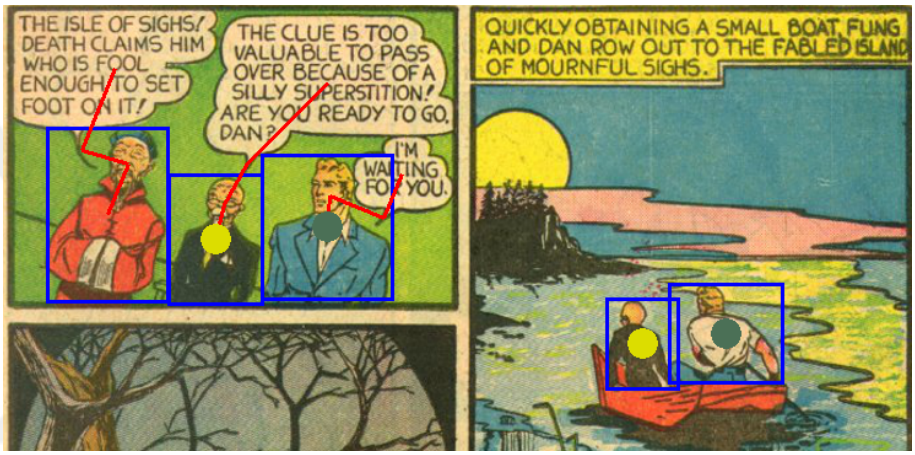

Figure 3.9: The UI when match mode is activated and the available matches are marked

The matching mode is activated when the `"I"` key is pressed. With the activation of this mode, only body boxes and association lines are made visible to the user. To annotate matches, the user must click the body boxes of the same character in the image. When the match process is done for one character, pressing `"N"` is mandatory to proceed with the next one. The matches are shown with colored dots in the centers of the body boxes. The matched characters have the same colored circles in the center of their body boxes. Figure 3.7 demonstrates an example where a list of matches is marked.

3.7.5 Text Box Transcription

The transcription mode is activated when the `"T"` key is pressed. Only speech bubbles and narrative boxes are visible to the user in this mode. If the user wants to enter the transcript of a text box, they just click on this box, and a slot becomes visible in the bottom-left corner of the UI to enter the transcript. Figure 3.10 is a snapshot of the UI in this mode after one of the text boxes is clicked on and its transcript is written.

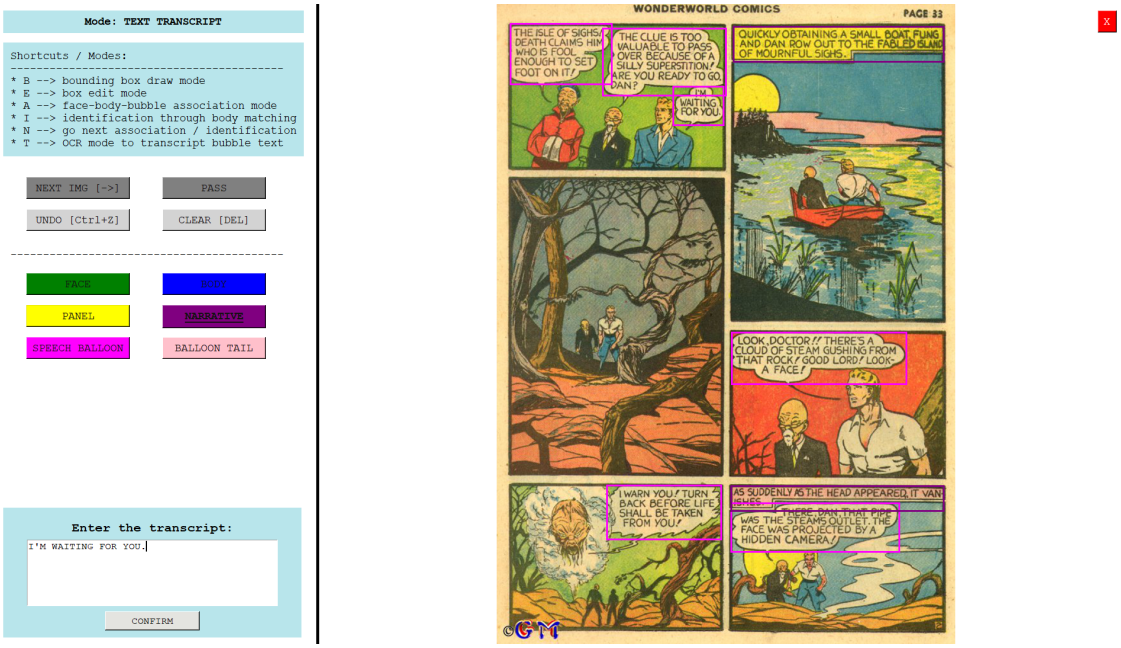

Figure 3.10: The UI when transcription mode is activated and one of the speech bubbles is clicked on

3.7.6 Saving Format

After all of the labelings are done, the user clicks on the `"NEXT IMG"` button, and the annotations are saved to a ”txt” file that has the same name as the original image. The format of the saved annotations is given below:

```
### BOXES
id-1,x1,y1,x2,y2,object_type
id-2,x1,y1,x2,y2,object_type
...
id-n,x1,y1,x2,y2,object_type

### ASSOCIATIONS
id-1,id-3,id-10
id-4,id-8
id-2,id-5,id-6,id-7
...

### IDENTIFICATIONS
id-1,id-4
id-2,id-12,id-20,id-16
...

### TEXTS
id-10,corresponding text
id-7,corresponding text
...
```

Each box object is assigned to an integer id, and all annotations are kept through these ids. `object_type` can be one of face, body, tail, bubble, narrative, or panel. There is no limit on the associated number of boxes in one association and also no limit on identification.

Chapter 4

RESULTS & DISCUSSION

In the following parts, I will explain my training details, discuss the effect of style transferring in stage 1, analyze the experiments done by utilizing the teacher-student network, and present my results retrieved after fine-tuning with limited and unlimited drawing data. Moreover, I will also state my outputs from the face generation and reconstruction tasks. I will use abbreviations [1] of datasets in the given tables to save space since there are many datasets for evaluation. Average Precision (AP) is selected as the evaluation metric for detection, and the intersection of union value for evaluation is fixed at 0.5.

4.1 Training Details

In all variations and experiments on detection, the batch size is set to 16, and one Tesla T4 GPU is used. AP scores are calculated by running the same variation five times and computing the average of these runs.

At stages 1 and 3, the learning rate is fixed at 0.001. The highest-scoring checkpoints in the evaluation set among 350 epochs are chosen as the final models. The first and the last 15 epochs include no augmentation. Otherwise, horizontal & vertical flips, the color distortion between $[-20°, 20°]$ degrees, shear, and mosaic augmentation (i.e., combining four random images and passing them as a single image) are applied randomly between the 15th and 335th epochs.

For the teacher-student network, the learning rate is set as 0.0001, and the best checkpoints in 10000 iterations are taken as final models. While the input image

[1] iCartoonFace as iCF, Manga109 faces as M109-F, Manga109 bodies as M109-B, DCM772 faces as DCM-F, DCM772 bodies as DCM-B, Comic2k as C2k, Watercolor2k as W2k, Clipart1k as C1k, and eBDtheque as eBD

of the teacher network is only horizontally flipped, Gaussian noise, color distortion, and random crop are applied additionally to the student network in all epochs.

On generation, I fix the batch size to 16 and use two Tesla T4 GPUs for StyleGAN2 and four for StyleGAN3-R. Default parameters for FFHQ are selected to train the models. While StyleGAN2 is trained for 4680 kimgs, StyleGAN3 is trained for 14000 kimgs. HFGI is utilized for reconstruction, the batch size is set as 8, and training is executed on one Tesla T4 GPU.

4.2 Style Transferred Pre-training

In this stage, I try to find the best combination to initialize the teacher-student network. For this purpose, I train the model variations with cartoonized natural images but evaluate them with drawing datasets. Scores retrieved after pre-training stage 1 are given in Table 4.1 for the individual styles and other experiments.

In the drawing domain, characters can be drawn in various styles. Although texture and colors continuously change among products, key fragments of faces and bodies preserve their existence (e.g., faces include at least one eye, and bodies contain either a head, arms, or legs). In my case, I believe that using multiple styles instead of one forces the model to focus more on to shape of the object rather than texture. Consequently, the model learns more generalizable information rather than style-specific; the objects are detected more accurately when the model is tested with unseen examples.

While leveraging even a single style transferring method from top-5 (i.e., *White-box, Hosoda, KH, Hayao* and *Shinkai*) ensures performance increase compared to using no styles, *All Styles* outperforms both individual styles and *Best Styles* (i.e., utilizing only top-5 performing styles), since increasing the number of style transferring options draws the model's focus on the shape more. Furthermore, adding animal annotations to the ground truth during the style transferred pre-training stage pushes the performance even further in datasets where animal-like character bodies are annotated (e.g., Manga109, DCM 772, and eBDtheque). However, *No Animals* model surpasses the *All Styles* in Watercolor 2k and Clipart 1k since this

Table 4.1: AP performances of my model after pre-training stage 1 for different style transferring variations: with single style transferring variation selected, combining all variations, combining top-5 best-performing variations, and without including animal annotations while combining all styles. Best score per dataset is colored with red, second with blue, third with green. The score is underlined if it is the best score among individual styles

Styles	iCF	M109-F	DCM-F	M109-B	DCM-B	C2k	W2k	C1k	eBD
Hayao	36.53 ± 0.77	42.34 ± 3.44	63.43 ± 1.44	28.25 ± 3.88	51.63 ± 4.48	47.88 ± 1.11	61.38 ±0.97	49.07 ±1.28	11.29 ±0.65
Shinkai	34.88 ± 1.26	41.21 ± 2.64	57.56 ± 3.11	30.84 ± 2.01	56.40 ± 1.85	48.21 ± 1.15	60.26 ±1.21	50.47 ±0.65	11.73 ±0.85
Hosoda	38.81 ± 0.40	49.59 ± 0.85	60.35 ± 3.13	36.50 ± 1.07	54.63 ± 3.13	51.12 ± 0.90	62.52 ±0.67	53.58 ±1.22	11.10 ±0.39
Paprika	32.27 ± 0.66	32.16 ± 4.17	50.95 ± 3.39	21.69 ± 1.84	40.40 ± 3.09	40.50 ± 1.64	47.96 ±2.43	40.46 ±1.58	7.84 ±0.50
Van Gogh	33.31 ± 1.66	35.30 ± 1.80	62.73 ± 1.30	26.09 ± 2.42	50.80 ± 2.56	44.66 ± 1.87	58.38 ±1.16	46.18 ±1.68	9.23 ±0.72
Monet	26.25 ± 2.98	30.44 ± 3.06	58.89 ± 4.06	21.13 ± 2.98	50.96 ± 1.54	39.31 ± 1.81	58.84 ±1.43	44.62 ±2.13	7.29 ±0.83
Cezanne	29.96 ± 0.76	35.49 ± 2.64	59.10 ± 2.59	26.76 ± 3.05	46.22 ± 8.25	41.50 ± 4.16	52.04 ±8.21	42.12 ±4.08	9.57 ±0.78
Miyazaki	32.16 ± 1.62	38.39 ± 0.94	59.63 ± 2.52	28.31 ± 1.53	55.91 ± 2.23	42.78 ± 0.29	61.31 ±0.45	47.83 ±1.71	10.18 ±0.88
AS	35.34 ± 0.94	39.81 ± 2.83	57.09 ± 0.66	27.01 ± 1.34	52.23 ± 2.19	44.79 ± 1.81	60.81 ±0.64	47.34 ±0.69	9.85 ±0.77
KH	37.69 ± 0.62	44.18 ± 2.20	59.68 ± 2.97	28.59 ± 0.65	49.10 ± 1.41	49.04 ± 1.14	63.51 ±1.00	50.34 ±0.73	11.69 ±0.61
Whitebox	42.22 ± 1.49	53.10 ± 2.12	59.63 ± 3.93	38.61 ± 1.73	52.46 ± 2.23	52.41 ± 1.25	63.35 ±0.41	54.56 ±1.64	12.52 ±1.31
No Styles	33.00 ± 1.97	40.44 ± 3.04	59.63 ± 3.83	30.69 ± 2.59	58.94 ± 3.75	44.81 ± 1.05	61.88 ± 0.95	49.60 ± 1.20	12.00 ± 1.24
All Styles	42.50 ± 1.25	54.74 ± 2.20	69.93 ± 2.67	42.72 ± 3.00	65.46 ± 1.35	56.80 ± 1.42	67.36 ± 0.39	55.65 ± 0.87	14.70 ± 0.95
No Animals	42.31 ± 0.70	52.09 ± 1.73	69.70 ± 2.26	37.53 ± 0.87	62.85 ± 1.16	54.58 ± 0.51	67.97 ± 0.24	58.37 ± 0.64	13.37 ± 0.58
Best Styles	42.04 ± 1.41	53.82 ± 3.28	65.94 ± 2.71	41.98 ± 1.94	59.96 ± 1.82	55.36 ± 1.56	66.91 ± 0.89	55.99 ± 0.73	14.25 ± 0.57

dataset consists of natural human-like characters rather than non-human appearances.

Although cartoonization methods convert natural images to drawing-like images, they also cause deterioration of fine details in the images. For the cases where variations result in a worse performance than the non-style-transferred version (i.e., *No Styles*), I observe that these variations are not only inferior in the cartoonization of humans but also cause a decrease in the quality of the image. Thus, utilizing only these variations provides a worse score than no style transferring.

4.3 Self-supervised Training

Unlike previous works, I pre-train my model with cartoonized natural images and execute the teacher-student stage with unlabeled drawing images. Because of the shift in the input domain, my initial teacher model remains more prone to FP and FN cases. To overcome this issue, I modify my loss function and student network update policy.

I extensively analyze the effect of each individual hyper-parameter set for the self-supervised teacher-student network. In this Section, I will discuss all of my experiments in this self-supervised stage. I will refer to Table 4.2 for the additional student network (SN) update interval (Φ), loss selection, EMA keep rate (d), regression loss coefficient (β), positive (c_{pos}^{thold}) and negative (c_{neg}^{thold}) student network confidence thresholds, and teacher network threshold (c_{teac}). Table 4.3 will be used to evaluate the optimizer selection, and Table 4.4 will be investigated to highlight the importance of style transferring before the self-supervised stage.

4.3.1 Loss Function

In experiments 2, 6, 7, and 8, my modified OHEM loss is compared with the SimOTA loss, which is the default loss method in YOLOX and an advanced variation of Focal loss. Es explained in Section 3.3, I believe that selecting a subset of predictions for backpropagation reduces the amount of misleading in FP and FN cases. My results also validate that OHEM loss is more suitable for my self-supervised architecture. Independent from the Φ value, models with OHEM loss outperform others with an average $\sim 1.5 - 2$ AP difference.

4.3.2 Updating SN per Φ Iterations

Between experiments 1 and 6, I try various iteration counts for Φ. Although this parameter is not too sensitive to different values, I observe that the overall performance drops if $\Phi > 500$. The score is worst if there is no manual SN update (i.e., $\Phi = None$). In Figure 4.1, the AP scores of the student and teacher networks per iteration are given for iCartoonFace, Manga109, and Comic2k while the Φ values

Table 4.2: AP scores of different unsupervised experiment configurations. Φ is the number of iterations where teacher weights are loaded to student networks afterward, d is the EMA keep rate, β is the coefficient of regression loss, c_{teac} is the confidence threshold of teacher network to select a prediction as pseudo ground truth, ct_{pos}^{thres} is the minimum confidence threshold for the student network prediction to be counted as positive in OHEM loss, ct_{neg}^{thres} is the maximum confidence threshold for the student network prediction to be counted as negative in OHEM loss. The AP Diff. column is calculated by averaging the maximum score in each dataset minus the experiment score. Best score per dataset is colored with red, second with blue, third with green. The models listed are selected from the checkpoints where they result in the best performance in average of all datasets

Index	Φ	Loss	d	β	ct_{pos}^{thres}	ct_{neg}^{thres}	c_{teac}	iCF	M109-F	DCM-F	M109-B	DCM-B	C2k	W2k	C1k	eBD	AP Diff.
1	250	OHEM	0.9996	2	0.15	0.85	0.65	49.10	69.14	81.52	69.28	77.52	67.13	71.18	64.63	25.20	0.59
2	500	OHEM	0.9996	2	0.15	0.85	0.65	49.05	69.23	82.22	69.41	77.83	67.38	71.60	64.12	25.22	**0.44**
3	1000	OHEM	0.9996	2	0.15	0.85	0.65	48.48	68.92	82.14	69.11	77.93	67.30	72.18	63.37	25.34	0.58
4	2000	OHEM	0.9996	2	0.15	0.85	0.65	48.71	68.79	81.91	67.30	77.50	66.79	72.07	63.83	24.85	0.91
5	5000	OHEM	0.9996	2	0.15	0.85	0.65	48.26	68.55	81.62	67.04	77.30	66.88	72.37	63.23	24.90	1.09
6	Never	OHEM	0.9996	2	0.15	0.85	0.65	47.83	68.32	81.71	67.03	77.29	67.21	72.33	63.27	24.77	1.14
7	500	SimOTA	0.9996	2	-	-	0.65	47.13	67.65	82.23	63.62	75.42	65.86	72.55	61.41	20.83	2.59
8	Never	SimOTA	0.9996	2	-	-	0.65	47.10	67.58	82.19	63.83	75.48	65.96	72.59	61.36	20.86	2.56
9	500	OHEM	0.9990	2	0.15	0.85	0.65	49.01	69.10	82.21	69.48	77.89	67.24	71.30	64.21	25.28	0.47
10	500	OHEM	0.9992	2	0.15	0.85	0.65	49.05	69.07	81.95	69.29	77.83	67.25	71.46	64.08	25.19	0.54
11	500	OHEM	0.9998	2	0.15	0.85	0.65	49.31	69.32	81.74	68.37	77.44	67.23	71.83	64.40	24.90	0.61
12	500	OHEM	0.9999	2	0.15	0.85	0.65	49.81	68.31	79.28	63.52	75.35	65.67	72.17	64.37	22.93	2.06
13	500	OHEM	0.9996	0	0.15	0.85	0.65	48.07	68.71	81.65	69.13	77.60	67.02	71.41	64.79	25.35	0.70
14	500	OHEM	0.9996	1	0.15	0.85	0.65	49.29	69.36	81.64	68.65	77.50	66.97	71.75	65.02	25.00	0.53
15	500	OHEM	0.9996	4	0.15	0.85	0.65	48.82	69.04	82.44	69.65	77.82	67.55	71.69	63.50	25.42	0.45
16	500	OHEM	0.9996	10	0.15	0.85	0.65	48.94	69.07	81.71	68.43	77.87	67.31	72.43	63.10	24.88	0.69
17	500	OHEM	0.9996	2	$0.5 \rightarrow 0.15$	$0.5 \rightarrow 0.85$	0.65	47.52	68.49	82.04	69.46	78.04	67.19	72.40	64.14	25.53	0.58
18	500	OHEM	0.9996	2	0.70	0.30	0.65	49.14	69.26	82.13	69.14	77.63	67.41	71.67	64.07	25.05	0.50
19	500	OHEM	0.9996	2	0.50	0.50	0.65	49.19	69.25	82.45	69.38	77.90	67.41	71.53	64.25	25.24	**0.38**
20	500	OHEM	0.9996	2	0.30	0.70	0.65	49.09	69.15	82.08	69.48	77.90	67.37	71.68	64.09	25.26	**0.43**
21	500	OHEM	0.9996	2	0.05	0.95	0.65	48.99	69.14	82.33	69.32	78.03	67.31	71.62	63.99	25.19	0.45
22	500	OHEM	0.9996	2	0.00	1.00	0.65	49.22	69.27	82.12	69.12	77.75	67.40	71.72	64.33	25.13	**0.44**
23	500	OHEM	0.9996	2	0.15	0.85	0.35	49.09	69.04	81.70	69.25	77.63	67.27	71.83	64.10	25.31	0.53
24	500	OHEM	0.9996	2	0.15	0.85	0.50	49.15	69.24	81.50	69.19	77.75	67.45	71.82	64.55	25.20	0.46
25	500	OHEM	0.9996	2	0.15	0.85	0.75	49.08	69.26	82.51	69.19	77.85	67.26	71.43	64.13	25.14	0.46
26	500	OHEM	0.9996	2	0.15	0.85	0.90	48.92	69.20	82.76	68.95	77.93	66.93	71.04	63.67	25.28	0.59

equal to *500* and *None*. By setting Φ to *500*, I prevent early performance drops in the student network, and this causes a more stable teacher development compared to $\Phi = None$. With this design, I improve my performance up to $\sim 1.5\%$ in most datasets.

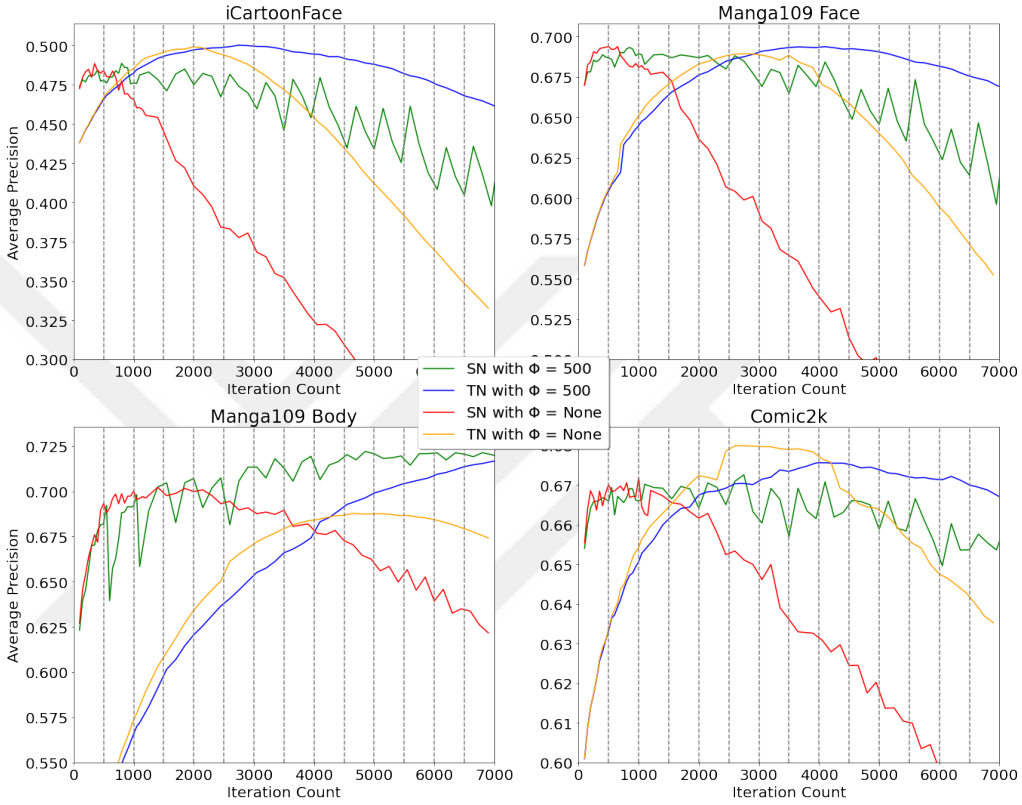

Figure 4.1: AP curves of teacher and student networks when Φ is 500 and None

4.3.3 Regression Coefficient β

Unlike the Unbiased Teacher [Liu et al., 2021], I have seen that using Smooth L1 regression loss in the self-supervised stage improves the scores further. The logic behind the teacher-student network is that the teacher module provides more accurate outputs than the student network due to the augmentation difference. Therefore, although the teacher network may not offer qualitative box regions, it should still predict more accurate areas than the student network. The student network should improve itself using the teacher network's predictions. In experiments 2 and 13-16,

the effect of β is demonstrated. $\beta \in \{2, 4\}$ are the most suitable values in this part of the pipeline.

4.3.4 EMA Keep Rate (d)

In the previous self-supervised detection studies, it has been shown that rates below 0.999 result in a worse performance. Therefore, I limit my rate range to $d \in \{0.9990, 0.9992, 0.9996, 0.9998, 0.9999\}$. Experiments 2, 9, 10, 11, and 12 contain the model performances with different d values. While I achieve the greatest average performance with $d = 0.9996$, I obtain similar scores with all values except 0.9999.

4.3.5 Student Confidence Thresholds(c_{pos}^{thold} and c_{neg}^{thold})

I test the influence of positive and negative SN confidence thresholds in experiments 2 and 17-22. With a dynamically changing threshold (exp. 17), or a threshold starting from too high for positive and too low for negative (exp. 18), the average performance is lower than the others. While the original OHEM loss corresponds to exp. 22, adding additional thresholds for SN results in greater or equal scores (e.g., experiments 2, 19, and 20). The best performance is obtained by setting both c_{pos}^{thold} and c_{neg}^{thold} to 0.5.

4.3.6 Teacher Confidence Threshold (c_{teac})

Stage 2 performance may also significantly change based on different confidence limits for selecting a TN prediction as a pseudo-ground-truth box. Between 2th and 23-26th experiments, I analyze how different teacher confidence thresholds change the AP result. Among the 5 values I test, $c_{teac} = 0.65$ gives the best average AP score among the datasets I evaluate. The outcomes are slightly worse, with the values smaller or larger than 0.65. However, the model is not too sensitive to this threshold since the *AP Difference* is only 0.02 for 0.5 and 0.7.

4.3.7 Optimizer Selection

My study states that manually changing SN's weights with TN's may mislead the overall model if an optimizer with momentum is utilized. To test my statement, I train two models with the same hyper-parameter configurations but select standard SGD in one and Nesterov SGD in the other (see Table 4.3). In almost every dataset, standard SGD scores $\sim 1.5 - 2\%$ higher. Especially in Manga 109 bodies, the difference increases to $\sim 5\%$. Hence, standard SGD is more suitable for my design.

Table 4.3: AP scores when standard SGD and Nesterov SGD are applied for optimization during self-supervised pre-training to the model variation with the hyper-parameters: $\Phi = 500$, $d = 0.9996$, $\beta = 2$, $c_{pos}^{thold} = 0.5$, $c_{neg}^{thold} = 0.5$, $c_{teac} = 0.65$. The best score per dataset is underlined

Optimizer	**iCF**	**M109-F**	**DCM-F**	**M109-B**	**DCM-B**	**C2k**	**W2k**	**C1k**	**eBD**
SGD	<u>49.19</u>	<u>69.25</u>	<u>82.45</u>	<u>69.38</u>	<u>77.90</u>	<u>67.41</u>	71.53	<u>64.25</u>	<u>25.24</u>
Nesterov SGD	48.72	67.26	80.10	64.17	76.59	66.81	<u>72.51</u>	62.20	23.49

4.3.8 Style Transferring Before Self-supervised Stage 2

In this part, I investigate if style transferring is needed in stage 1 before applying self-supervised stage 2. As in the *Optimizer Selection*, I train two models with the same settings but initialize the pre-trained weights of these models in the teacher-student stage differently: one with the weights retrieved from pre-training stage 1, including style transferring, the other without style transferring. Table 4.4 shows the results from these experiments. While the AP difference is smaller in body evaluations ($\sim 2 - 3\%$), it increases up to $\sim 10\%$ in face data. Either way, applying style transferring in stage 1 has a significant positive effect on the self-supervised stage 2 model performance.

Table 4.4: AP scores after self-supervised pre-training stage 2 when style transferring is applied or not during pre-training stage 1. The hyper-parameters for the model in this table are: $\Phi = 500$, $d = 0.9996$, $\beta = 2$, $c_{pos}^{thold} = 0.15$, $c_{neg}^{thold} = 0.85$, $c_{teac} = 0.65$. The best score per dataset is underlined

Style Transfer	**iCF**	**M109-F**	**DCM-F**	**M109-B**	**DCM-B**	**C2k**	**W2k**	**C1k**	**eBD**
Yes	49.05	69.23	82.22	69.41	77.83	67.38	71.60	64.12	25.22
No	41.66	58.96	74.56	66.32	75.64	65.32	69.56	64.13	25.16

4.3.9 Comparison with SOTA and Discussion

In Table 4.5, I compared my model with the previous supervised and weakly-supervised SOTA detectors on drawings. Although my teacher-student (TS) model does not leverage any drawing data annotation, it outperforms the previous supervised SOTA in DCM 772, in which the drawing style is more like natural images. Furthermore, it surpasses the previous weakly supervised SOTA in Manga 109, DCM 772, Comic2k, and Clipart 1k. On the other hand, if the stylistic variety between the labeled cartoonized training data and testing data increases (e.g., iCartoonFace and Manga 109), then the performance gap between supervised SOTA detectors and my TS model broadens as well. This issue is caused by my teacher model identifying the correct face and body areas in the drawings but failing to recognize the characters with appearances significantly different from the style transferred natural images I used for pre-training. Therefore, pseudo-labels cannot be created for these characters, and the model cannot train itself for these cases. The performance gap is especially large in iCartoonFace since this dataset maintains many non-human-like characters, but my cartoonized training data includes annotations for only human faces. Still, the TS model preserves its stability on different datasets.

Table 4.5: Overall AP performances of my models and previous SOTA models. The teacher-student network is initialized with the style transferred pre-training, all of my supervised models are initialized with pre-training stage 2 weights. NS: no target domain supervision. SS: self-supervision, WS: weak-supervision, FS: full target domain supervision. Best score per dataset is colored with red, second with blue, scores with "*" mean that they are evaluated by me using the model from the original project repository. "**" indicates that the results are retrieved from single-dataset trainings and each score is calculated by a separate model trained specifically with the particular dataset

Types	Models	iCF	M109-F	DCM-F	M109-B	DCM-B	C2k	W2k	C1k	eBD
NS	All Styles	42.50 ± 1.25	54.74 ± 2.20	69.93 ± 2.67	42.72 ± 3.00	65.46 ± 1.35	56.80 ± 1.42	67.36 ± 0.39	55.65 ± 0.87	14.70 ± 0.95
SS	Teacher-Student	49.05 ± 0.15	69.23 ± 0.15	82.22 ± 0.40	69.41 ± 0.40	77.83 ± 0.19	67.38 ± 0.18	71.60 ± 0.10	64.12 ± 0.13	25.22 ± 0.14
WS	Inoue et al. [Inoue et al., 2018]	-	-	-	36.71*	41.89*	57.30	73.20	63.00	28.20*
SS	UMT [Deng et al., 2020]	-	-	-	-	-	-	69.90	70.50	-
SS	D-adapt [Jiang et al., 2021]	-	-	-	-	-	53.50	68.90	69.30	-
FS	Train w/ 64 Images **	65.47 ± 0.67	80.41 ± 1.41	69.80 ± 3.38	77.72 ± 0.74	77.28 ± 1.45	68.36 ± 1.43	71.24 ± 1.22	58.74 ± 2.57	-
FS	Train w/ 512 Images **	74.39 ± 0.60	85.15 ± 0.22	74.85 ± 0.28	82.32 ± 0.24	82.40 ± 0.57	71.05 ± 0.94	77.63 ± 1.08	71.72 ± 1.55	-
FS	Train w/ All Images **	87.75 ± 0.02	87.86 ± 0.02	75.87 ± 2.79	87.06 ± 0.10	84.89 ± 0.20	71.66 ± 0.37	89.17 ± 0.16	77.97 ± 0.35	-
FS	ACFD [Zhang et al., 2020a]	90.94	-	-	-	-	-	-	-	-
FS	Ogawa et al. [Ogawa et al., 2018]	-	76.20	-	79.60	-	-	-	-	-
FS	Nguyen et al. [Nguyen et al., 2018]	-	-	74.94	-	76.76	-	-	-	-
FS	Inoue et al. [Inoue et al., 2018]	-	-	-	-	-	70.10	77.30	76.20	-

4.4 Supervised Training

I train my architecture for single datasets with limited instances to evaluate their behavior when only a low amount of data is available. The average of scores for all datasets (i.e., iCartoonFace, Manga 109, DCM 772, Comic2k, Watercolor2, Clipart 1k) are shared in Table 4.6. In the cases with extremely low instance counts (i.e., 64 and 128 images), utilization of natural images and self-supervised learning results in up to $\sim$ 24% performance increase compared to starting from a random initial state. When trained with all available data, both style-transferring-based and teacher-student-based pre-training methods score similar values. I believe this is caused since there is sufficient data for these specific sub-domains to close the gap that emerged

from the self-supervised stage. However, I still obtain a significant improvement (∼ 1.2%) when models start from pre-trained weights instead of random initialization. This shows that leveraging style transferred pre-training enhances the performance independently from the amount of labeled fine-tuning data.

Table 4.6: Average AP performance of my model when trained with a subset of individual datasets having annotations of a limited number of random images. Best scores per image instance count are colored with red

	Image Instance Counts					
Initial Weights	**64**	**128**	**256**	**512**	**1024**	**All**
Random Initialization	47.79 ± 1.38	56.83 ± 1.31	63.57 ± 0.74	69.38 ± 0.82	75.64 ± 1.49	80.60 ± 0.65
After Pre-training w/ Style Transfer	66.90 ± 1.40	70.51 ± 1.50	73.75 ± 0.55	75.34 ± 0.99	79.38 ± 1.04	82.87 ± 1.53
After Teacher-Student	71.13 ± 0.92	73.36 ± 0.70	74.98 ± 0.79	77.44 ± 0.47	79.38 ± 0.23	82.78 ± 0.93

In Table 4.5, I compare previous SOTA models with my best results from each stage checkpoint (i.e., style transfer, teacher-student, fine-tuning with individual datasets). My model achieves close scores to ACFD [Zhang et al., 2020a] and outperforms the Cross-Domain model [Inoue et al., 2018]. Even with a low amount of training images, I obtain better or comparable results with DCM 772 model of Nguyen et al. [Nguyen et al., 2018] and Manga 109 model of Ogawa et al. [Ogawa et al., 2018].

4.5 Face Generation & Reconstruction

This section shares metric and visual results on face generation and reconstruction tasks. Please note that this task is completed as an additional task to measure generation performance on drawings if a large-scale dataset is available. Since there is no study about face generation on the *Comic Faces in the Wild* dataset or drawings with such a large stylistic range, I cannot provide any other study compared to my results. Additionally, StyleGAN2 could not be trained as many iterations as StyleGAN3-R due to time limitations. The results shared in the following subsec-

tions are only preliminary outcomes of the possible future works on image generation that can be conducted on drawings.

4.5.1 Metric Results

Table 4.7: Generation scores of each trained GAN model. Lower is better in FID and KID. Higher is better in Precision and Recall

Model	**Resolution**	**kimgs**	**FID 50k**	**KID 50k**	**Precision 50k**	**Recall 50k**
StyleGAN2	256 x 256	4680	16.3215	0.0085	0.6532	0.2346
StyleGAN3-R	128 x 128	14000	6.3177	0.0033	0.6133	0.4718
StyleGAN3-R	256 x 256	14000	10.3709	0.0055	0.6205	0.3726

Four metrics are chosen to measure the generation performance: Fretchet Inception Distance (FID), Kernel Inception Distance (KID), Precision, and Recall. Lower is better in FID and KID; higher is better in Precision and Recall. These scores are calculated through 50,000 randomly generated images from the GAN models. The results are given in Table 4.7. Decreasing the output resolution causes a better performance on the same number of kimgs. The best scores are retrieved from StyleGAN3-R with 128 x 128 resolution, which is trained on 4 Tesla T4 GPUs for ∼12 days. Original StyleGAN3-R continues the training until 25,000 kimgs in FFHQ. Therefore, increasing the training time may further improve performance.

Table 4.8: Reconstruction scores of HFGI model with StyleGAN2 backbone. Lower is better in both LPIPS and MSE

Model	**Resolution**	**kimgs**	**LPIPS**	**MSE**
HFGI	256 x 256	480	0.0778	0.0170

On reconstruction, final StyleGAN2 with 256 x 256 resolution is utilized since HFGI currently does not support StyleGAN3-R. HFGI is trained on 480 kimgs and evaluated with two metrics: Learned Perceptual Image Patch Similarity (LPIPS) and Mean Squared Error (MSE). Table 4.8 shows the results.

4.5.2 Visual Results

Figures 4.2 and 4.3 provide randomly generated drawing faces from StyleGAN3-R and StyleGAN2 models. Both of these models capture a wide range of stylistic variations such as anime vs. western, colored vs. black-white, old comics vs. new comics, human vs. non-human characters, gender, facial orientation, masked vs. not masked faces. However, unequal or missing facial organs and glitches in images are still clearly visible. Furthermore, since the face images from the crawled data and COMICS are significantly dominant, the majority of the generated faces are more similar to these styles (e.g., fewer black-white or anime-like faces). To overcome this issue, the data should be cleaned, and the stylistic distributions should be equalized. However, these tasks are left as future work since the main focus of this thesis is not face generation or dataset creation.

Figure 4.4 shows the reconstruction results taken from HFGI. The model preserves main features of the given faces while reconstructing them. Using intermediate latent embeddings retrieved through this model may also include meaningful features for possible future tasks such as person recognition or emotion recognition.

Figure 4.2: Randomly generated 128 x 128 faces from the trained StyleGAN3-R

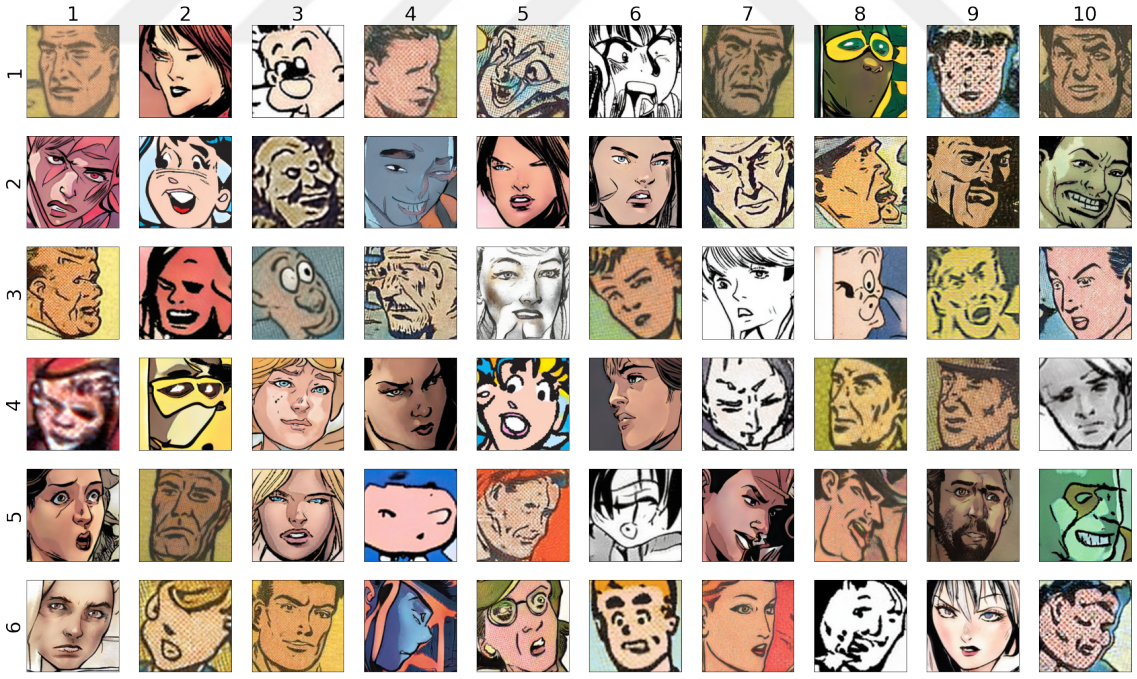

Figure 4.3: Randomly generated 256 x 256 faces from the trained StyleGAN2

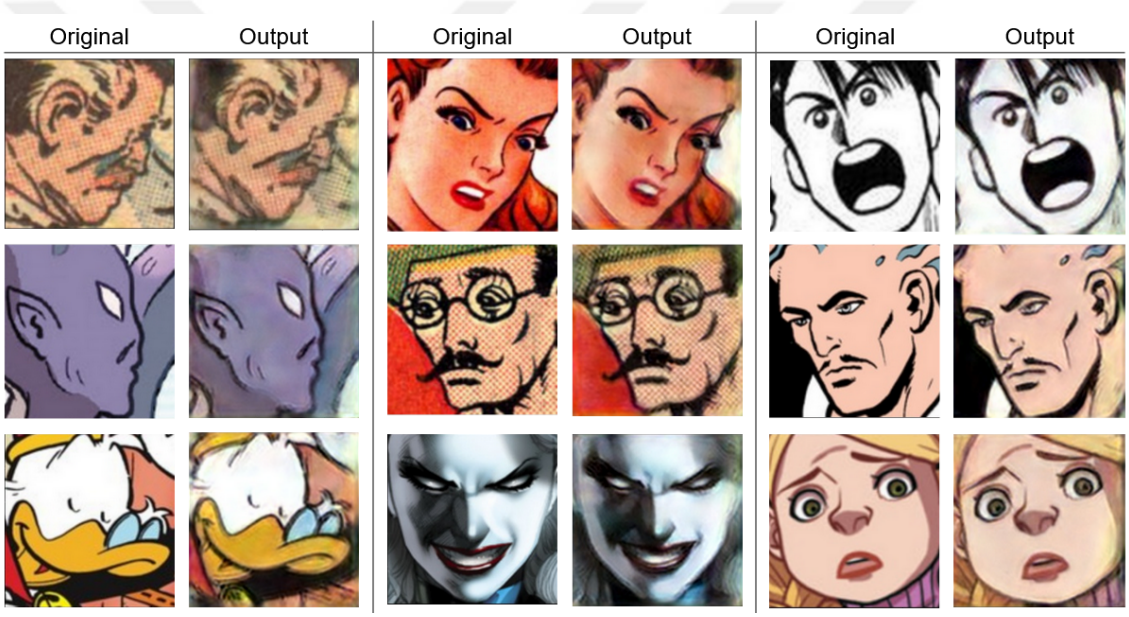

Figure 4.4: Randomly reconstructed 256 x 256 faces from the trained HFGI

Chapter 5

ABLATION STUDY

5.1 Stylistic Domain Coverage of Individual Datasets

In my study, I express that the labeled drawing data only covers a small subset of the overall domain in terms of stylistic variety. To analyze the stylistic coverage of individual datasets, I design an experiment where the models, that are fine-tuned on a single labeled dataset with a limited number of instances, are evaluated on the other annotated datasets (e.g., if my model is fine-tuned in iCartoonFace, then I also evaluate that model in Manga 109 and DCM 772 faces). By comparing these results, I can infer how valuable each dataset is in other unseen styles/sub-domains. To prevent my model from being affected by other datasets, I will base my statements on the randomly initialized fine-tuning in the Tables 5.1 and 5.2 (i.e., the columns titled with N) during my analysis if no other pre-training is mentioned.

In the following parts, I will use the notation $train\_data^{eval\_data}_{number\_of\_images}$ to mention the models and their results (e.g., if a model is trained in iCartoonFace with 64 images and evaluated on DCM faces, then the notation will be icf^{dcm}_{64}).

5.1.1 In Face Data

Regarding the scores on Table 5.1, I observe that my fine-tuned models with iCartoonFace and Manga 109 perform better in other datasets than the variations with DCM 772. In both iCartoonFace and Manga 109, when trained with one dataset (i.e., source sub-domain) and evaluated on the other (i.e., target sub-domain), using 1024 images from the source sub-domain is almost equal to utilizing 128 images from the target sub-domain for training (i.e., $icf^{m109}_{1024} \approx m109^{m109}_{128}$ and $m109^{icf}_{1024} \approx icf^{icf}_{128}$). If all instances are allowed to be leveraged, this equality changes to All & 512 for both. However, the models trained with DCM 772 obtain significantly worse

Table 5.1: AP performances of my model after stage 3 fine-tuning when trained with a subset of individual datasets having annotations of a limited number of random images. N: no pre-training, ST: pre-trained with style transferred images, SS: additional teacher-student pretraining. Underlined if the score is highest among all pre-training options for particular dataset and instance count, blue if the score of the evaluated dataset is higher than the previous supervised SOTA detector, and red if the score is the best for the particular dataset in this table

		Evaluation Datasets								
Training Datasets	**# of images**	**iCF**			**M109-F**			**DCM-F**		
		N	**ST**	**SS**	**N**	**ST**	**SS**	**N**	**ST**	**SS**
iCF	**64**	42.29 ± 5.18	61.67 ± 1.27	65.47 ± 0.67	36.40 ± 7.84	66.50 ± 3.68	73.55 ± 1.51	26.97 ± 5.77	52.40 ± 11.38	73.31 ± 3.35
iCF	**128**	52.18 ± 1.89	64.41 ± 1.05	68.58 ± 0.62	48.78 ± 5.57	69.00 ± 3.91	75.85 ± 1.06	31.13 ± 9.91	59.36 ± 7.81	72.35 ± 4.07
iCF	**256**	60.87 ± 0.68	69.20 ± 0.91	71.24 ± 0.65	62.25 ± 2.11	74.51 ± 1.88	76.08 ± 0.67	44.11 ± 4.78	60.94 ± 8.46	70.59 ± 1.85
iCF	**512**	66.22 ± 0.49	72.51 ± 1.38	74.39 ± 0.60	68.97 ± 3.51	75.38 ± 2.59	78.43 ± 1.46	50.91 ± 1.92	58.10 ± 9.41	66.30 ± 5.25
iCF	**1024**	72.47 ± 0.78	77.22 ± 2.13	77.31 ± 0.30	73.36 ± 2.74	78.36 ± 3.65	80.59 ± 0.47	53.44 ± 3.90	61.43 ± 3.84	67.90 ± 2.87
iCF	**All**	83.70 ± 0.21	87.61 ± 0.07	87.75 ± 0.02	83.33 ± 0.49	85.62 ± 0.06	85.63 ± 0.13	64.16 ± 0.98	71.98 ± 1.16	72.11 ± 0.41
M109	**64**	25.28 ± 4.19	47.36 ± 4.06	51.99 ± 3.04	67.46 ± 3.46	77.70 ± 4.05	80.41 ± 1.41	22.60 ± 8.16	54.40 ± 7.13	70.90 ± 3.70
M109	**128**	34.54 ± 3.29	49.40 ± 1.20	53.47 ± 1.18	74.89 ± 0.82	80.35 ± 0.54	82.14 ± 1.12	35.45 ± 5.18	56.63 ± 5.65	71.02 ± 1.44
M109	**256**	39.59 ± 3.96	53.11 ± 0.83	56.11 ± 1.97	76.80 ± 0.49	83.35 ± 0.62	84.20 ± 0.68	41.42 ± 10.71	57.83 ± 3.10	71.71 ± 3.12
M109	**512**	48.55 ± 1.68	58.16 ± 2.48	58.74 ± 1.42	82.98 ± 0.54	85.58 ± 0.61	85.15 ± 0.22	53.83 ± 4.76	63.56 ± 4.02	74.17 ± 1.86
M109	**1024**	51.41 ± 2.40	60.80 ± 0.95	62.46 ± 0.80	85.83 ± 0.41	86.50 ± 0.27	86.21 ± 0.19	57.74 ± 4.22	67.99 ± 2.21	72.87 ± 1.43
M109	**All**	66.84 ± 1.56	69.89 ± 0.65	70.71 ± 0.49	87.70 ± 0.05	87.87 ± 0.07	87.86 ± 0.02	77.20 ± 1.66	75.15 ± 1.36	78.40 ± 1.86
DCM	**64**	13.75 ± 2.13	32.43 ± 4.53	35.14 ± 2.56	32.28 ± 4.83	57.92 ± 4.59	62.15 ± 2.55	57.15 ± 2.20	66.64 ± 3.62	69.80 ± 3.38
DCM	**128**	19.63 ± 2.87	41.46 ± 3.37	38.38 ± 2.12	39.11 ± 3.06	64.38 ± 1.89	63.96 ± 1.44	67.01 ± 2.12	69.04 ± 5.26	73.78 ± 2.07
DCM	**256**	23.04 ± 1.41	36.56 ± 2.20	42.73 ± 0.95	49.00 ± 4.13	62.80 ± 3.14	68.25 ± 1.39	68.54 ± 0.93	75.34 ± 1.30	73.72 ± 2.35
DCM	**512**	29.13 ± 2.90	38.04 ± 2.73	43.93 ± 1.65	55.09 ± 2.46	67.02 ± 2.74	71.50 ± 1.38	71.76 ± 2.48	71.01 ± 2.95	74.85 ± 0.28
DCM	**1024**	34.78 ± 2.26	44.06 ± 2.09	46.70 ± 0.56	62.53 ± 2.23	71.73 ± 1.69	73.17 ± 0.67	74.66 ± 5.09	78.06 ± 2.89	75.93 ± 0.52
DCM	**All**	45.01 ± 1.67	47.22 ± 1.55	49.24 ± 0.22	68.17 ± 2.62	72.25 ± 1.01	73.26 ± 0.04	78.27 ± 0.32	79.48 ± 3.48	75.87 ± 2.79

performances in both iCartoonFace and Manga 109 (e.g., $dcm^{icf}_{All} < icf^{icf}_{128}$ and $dcm^{m109}_{All} \approx m109^{m109}_{64}$). Lastly, Manga 109 models outperform iCartoonFace models on DCM 772, if all of the training data are leveraged (i.e., $m109^{dcm}_{All} > icf^{dcm}_{All}$). In conclusion, while DCM 772 is the worst choice for the unseen sub-domains, both iCartoonFace and Manga 109 result similarly on the other. However, Manga 109 distinguishes further compared to iCartoonFace due to its greater score on DCM 772. The main reason for this difference may be the scope of the facial labeling in DCM 772. In DCM 772, only the faces of human-like characters are labeled. iCartoonFace includes significantly more non-human-like characters compared to Manga 109. Thus, detection of these non-human-like faces drops the performance on DCM 772.

5.1.2 In Body Data

If the training data is limited to at least 512 images, Manga 109 also distinguishes among the body datasets by outperforming Comic 2k* models ("*" explained in Table 5.2) on DCM 772 and DCM 772 models on Comic 2k & Watercolor 2k (e.g., $m109^{dcm}_{1024} > c2k*^{dcm}_{1024}$, $m109^{c2k}_{1024} > dcm^{c2k}_{1024}$, and $m109^{w2k}_{512} > dcm^{w2k}_{512}$). If DCM 772 models and Comic 2k* models are compared further on Manga 109 dataset, then DCM 772 models obtain greater scores if image instances are limited to 64-256. For the larger-size-limitation or no-limitation cases, the AP scores of Comic 2k* and DCM 772 models are similar. Unlike DCM 772's scope of facial labeling, animal bodies and background characters are annotated for bodies. Thus, the problem mentioned in the previous subsection does not apply to body detection.

5.2 Effect of Pre-training on Low and High Data

In this Section, I examine the effects of my pre-training strategies on stage 3 fine-tuning performance. I execute fine-tuning and evaluation on the same dataset and discuss the influence of both low and high amounts of training data.

While pre-training is always advantageous against random initialization, additional self-supervised pre-training outperforms the style transferred pre-training

Table 5.2: AP performances of my model after stage 3 fine-tuning when trained with a subset of individual datasets having annotations of a limited number of random images. C2k* indicates that all Comic2k, Watercolor2k, and Clipart1k datasets are combined for training the model. N: no pre-training, ST: pre-trained with style transferred images, SS: additional teacher-student pretraining. Underlined if the score is highest among all pre-training options for particular dataset and instance count, blue if the score of the evaluated dataset is higher than the previous supervised SOTA detector, and red if the score is the best for the particular dataset in this table

		Evaluation Datasets								
Training Datasets	# of images	M109-B			DCM-B			eBD		
		N	ST	SS	N	ST	SS	N	ST	SS
M109	64	54.36 ± 4.04	72.52 ± 1.07	77.72 ± 0.74	20.83 ± 5.11	66.94 ± 1.73	74.47 ± 0.38	6.04 ± 2.60	22.79 ± 2.26	22.25 ± 1.39
M109	128	64.18 ± 2.18	75.58 ± 1.25	79.27 ± 0.74	31.44 ± 6.17	66.96 ± 3.80	72.52 ± 2.75	8.53 ± 3.27	22.16 ± 2.65	22.29 ± 1.34
M109	256	71.68 ± 0.54	77.78 ± 0.52	80.79 ± 0.61	36.68 ± 4.07	64.30 ± 4.78	73.13 ± 0.28	12.69 ± 1.55	21.42 ± 2.58	23.55 ± 1.93
M109	512	75.37 ± 0.28	81.63 ± 1.25	82.32 ± 0.24	47.61 ± 2.15	69.46 ± 3.27	74.87 ± 1.20	17.57 ± 1.38	23.54 ± 2.51	25.61 ± 1.55
M109	1024	79.67 ± 1.48	83.32 ± 0.40	83.51 ± 0.35	53.46 ± 3.83	71.34 ± 2.07	76.89 ± 0.68	18.00 ± 1.45	25.36 ± 2.09	26.40 ± 1.07
M109	All	85.78 ± 0.22	87.15 ± 0.04	87.06 ± 0.10	71.27 ± 2.46	75.30 ± 1.13	78.06 ± 1.12	27.85 ± 0.55	29.78 ± 1.27	30.70 ± 1.16
DCM	64	37.51 ± 4.20	62.79 ± 3.71	68.90 ± 1.18	46.55 ± 1.78	71.97 ± 4.72	77.28 ± 1.45	10.19 ± 1.02	20.17 ± 2.65	22.67 ± 0.60
DCM	128	43.57 ± 3.77	66.92 ± 3.57	68.29 ± 1.47	54.80 ± 1.04	76.71 ± 1.94	78.34 ± 0.36	13.38 ± 1.05	23.31 ± 1.32	22.15 ± 0.57
DCM	256	53.20 ± 1.41	66.17 ± 2.76	71.40 ± 1.45	64.63 ± 1.22	79.27 ± 1.03	80.91 ± 1.59	15.66 ± 1.52	22.18 ± 0.75	24.45 ± 1.24
DCM	512	58.32 ± 4.07	70.29 ± 2.42	73.60 ± 0.47	69.93 ± 1.45	80.84 ± 1.44	82.40 ± 0.57	18.45 ± 1.06	25.54 ± 1.64	25.95 ± 0.80
DCM	1024	64.58 ± 0.86	72.00 ± 1.78	74.96 ± 0.51	76.89 ± 1.27	83.31 ± 0.58	83.81 ± 0.43	21.24 ± 1.31	24.63 ± 1.23	26.76 ± 0.41
DCM	All	68.01 ± 1.33	70.42 ± 2.17	70.09 ± 0.39	81.16 ± 0.29	84.66 ± 0.55	84.89 ± 0.20	28.19 ± 0.27	26.96 ± 1.33	28.08 ± 0.17
C2k*	64	31.04 ± 1.96	56.78 ± 4.99	67.94 ± 1.51	24.20 ± 3.34	58.17 ± 3.11	72.52 ± 0.84	6.62 ± 1.65	16.90 ± 3.84	20.47 ± 2.06
C2k*	128	40.02 ± 4.74	62.81 ± 2.46	69.64 ± 2.06	32.06 ± 5.57	65.39 ± 3.61	71.86 ± 2.24	7.35 ± 2.98	20.77 ± 2.60	20.26 ± 1.54
C2k*	256	50.43 ± 3.75	67.76 ± 2.48	72.13 ± 0.74	45.63 ± 1.50	67.12 ± 4.00	73.43 ± 1.48	12.06 ± 1.51	20.17 ± 1.83	21.64 ± 1.43
C2k*	512	58.60 ± 0.52	67.87 ± 1.52	72.53 ± 1.71	54.99 ± 2.37	69.26 ± 4.94	72.92 ± 1.24	16.34 ± 1.62	21.74 ± 1.40	23.09 ± 1.36
C2k*	1024	63.20 ± 1.60	70.78 ± 0.98	75.87 ± 0.41	59.03 ± 0.92	71.03 ± 1.44	76.84 ± 0.65	18.57 ± 0.79	22.42 ± 1.42	24.43 ± 0.29
C2k*	All	68.13 ± 1.84	71.81 ± 0.35	71.57 ± 0.19	67.78 ± 1.99	70.29 ± 0.67	71.10 ± 0.52	21.70 ± 2.22	22.73 ± 0.80	22.29 ± 0.15

Training Datasets	# of images	C2k			W2k			C1k		
		N	ST	SS	N	ST	SS	N	ST	SS
M109	64	32.12 ± 4.97	59.04 ± 2.25	66.13 ± 0.96	27.98 ± 8.40	60.11 ± 4.49	66.49 ± 1.49	11.87 ± 4.68	41.62 ± 3.44	47.63 ± 3.17
M109	128	43.23 ± 3.37	62.22 ± 1.18	65.36 ± 1.33	40.86 ± 4.20	61.96 ± 0.94	67.22 ± 1.66	16.36 ± 6.90	43.10 ± 4.20	47.81 ± 3.55
M109	256	49.97 ± 1.71	63.69 ± 1.48	65.85 ± 0.46	48.42 ± 3.05	62.65 ± 1.51	67.42 ± 1.12	24.32 ± 2.25	45.61 ± 2.56	48.86 ± 2.36
M109	512	58.18 ± 1.58	64.05 ± 2.32	66.86 ± 0.80	58.30 ± 1.31	65.69 ± 2.44	68.89 ± 1.06	33.77 ± 2.26	49.20 ± 2.66	51.76 ± 1.62
M109	1024	61.09 ± 0.87	65.85 ± 0.71	68.22 ± 0.70	62.30 ± 1.85	68.04 ± 0.85	70.75 ± 1.36	37.28 ± 3.73	52.01 ± 1.21	54.42 ± 3.17
M109	All	69.34 ± 0.79	71.60 ± 1.09	73.40 ± 0.36	70.68 ± 1.38	72.71 ± 1.04	73.77 ± 1.06	51.93 ± 1.70	57.60 ± 0.39	58.13 ± 0.56
DCM	64	37.97 ± 2.39	60.87 ± 3.04	65.51 ± 0.64	32.86 ± 3.59	61.44 ± 3.86	68.34 ± 0.96	16.32 ± 2.79	46.35 ± 5.53	54.29 ± 1.93
DCM	128	42.69 ± 0.85	62.90 ± 3.56	65.90 ± 0.47	37.96 ± 3.09	63.77 ± 4.41	69.11 ± 0.81	22.03 ± 1.79	51.62 ± 5.84	55.61 ± 1.07
DCM	256	50.63 ± 1.35	62.67 ± 1.86	66.98 ± 0.17	43.55 ± 3.99	61.18 ± 2.11	68.98 ± 0.50	30.49 ± 2.54	52.53 ± 1.86	58.58 ± 1.09
DCM	512	54.85 ± 2.37	64.64 ± 1.60	67.89 ± 0.43	50.61 ± 2.60	65.03 ± 1.36	70.54 ± 0.72	38.88 ± 1.73	54.56 ± 4.60	60.03 ± 1.06
DCM	1024	60.97 ± 0.41	66.29 ± 1.03	68.28 ± 0.17	57.38 ± 2.83	66.01 ± 1.80	70.89 ± 0.76	47.07 ± 1.05	58.26 ± 2.93	61.67 ± 1.58
DCM	All	65.11 ± 1.12	66.77 ± 0.73	66.83 ± 0.35	63.36 ± 1.53	66.85 ± 1.47	66.27 ± 0.83	54.00 ± 1.71	61.11 ± 1.82	62.43 ± 0.78
C2k*	64	41.42 ± 1.21	61.66 ± 1.87	68.36 ± 1.43	50.51 ± 1.64	66.90 ± 1.50	71.24 ± 1.22	22.57 ± 3.50	56.16 ± 3.01	58.74 ± 2.57
C2k*	128	50.39 ± 1.77	65.83 ± 1.14	69.10 ± 0.73	57.20 ± 3.76	69.66 ± 1.10	73.38 ± 1.30	33.96 ± 4.67	62.46 ± 2.42	62.27 ± 2.30
C2k*	256	57.79 ± 1.53	68.75 ± 0.68	69.96 ± 0.96	64.12 ± 2.49	72.25 ± 0.43	73.83 ± 1.04	44.13 ± 2.08	64.06 ± 2.12	65.18 ± 2.60
C2k*	512	63.43 ± 1.37	68.89 ± 0.26	71.05 ± 0.94	72.69 ± 2.26	74.47 ± 0.66	77.63 ± 1.08	52.67 ± 1.61	67.79 ± 2.79	71.72 ± 1.55
C2k*	1024	67.76 ± 0.79	70.63 ± 0.43	73.81 ± 0.21	83.38 ± 1.18	83.68 ± 2.42	79.63 ± 0.86	64.45 ± 1.04	72.33 ± 1.37	74.83 ± 0.61
C2k*	All	69.85 ± 1.15	72.16 ± 0.41	71.66 ± 0.37	86.20 ± 0.79	88.84 ± 0.70	89.17 ± 0.16	68.87 ± 2.71	77.52 ± 0.84	77.97 ± 0.35

mainly on the small size of training data. Additionally, with my pre-training design, I achieve higher scores than previous supervised SOTA models, even with few images. In Tables 5.1 and 5.2, I shared the model evaluations after fine-tuning with single datasets and a limited number of instances. The scores are written in blue if they exceed the previous supervised SOTA drawing detectors.

Self-supervised pre-training before fine-tuning improves detection performance, especially in low amounts of data, but it is still beneficial for higher data sizes compared to the random initialization. When trained with only 64 images, there is up to $\sim 7\%$ difference between initializing with style transferred stage 1 weights and self-supervised stage 2 weights. The difference increases up to $\sim 35\%$ against random initialization. On the other hand, if the size of data increases, the margin between different initialization methods decreases. But still, the performance with random initialization on no data limitation is approximately $\sim 2\%$ worse than with self-supervised stage 2 and style transferred stage 1. This indicates that pre-training is essential for higher performance even with a high amount of data.

I obtain better performances than most previous supervised SOTAs, even with tiny subsets of the datasets during fine-tuning. 256 panels from Manga 109 are enough to outperform the previous SOTA [Ogawa et al., 2018]. This changes to 1024 panels in DCM 772 [Nguyen et al., 2018], 512 images in Comic 2k and Watercolor 2k [Inoue et al., 2018]. I achieve better results when I utilize all data in Clipart 1k but fail to pass the iCartoonFace SOTA. While I aim to keep the model size and design simpler, the current iCartoonFace SOTA (ACFD [Zhang et al., 2020a]) is $\times 5$ times larger compared to my model and specifically designed for the iCartoonFace challenge. Still, my final performance is only $\sim 3\%$ smaller.

5.3 Increasing the Model Size

Until this part, all my experiments are conducted with the tiniest version of the YOLOX architecture. Therefore, I decide to measure the performance of my pipeline to the largest YOLOX architecture as well if all of the available annotated data are used. Following the same pipeline I propose until this section, I pre-train the largest

version of YOLOX and fine-tune individually on the labeled datasets. The results can be seen in Table 5.3. By increasing the model size, my scores are also advancing further. Especially the gap between iCartoonFace SOTA and my model decreases to 0.93 AP.

Table 5.3: Overall AP performances of my supervised models and previous SOTA models. Best score per dataset is colored with red, second with blue

Models	**iCF**	**M109-F**	**DCM-F**	**M109-B**	**DCM-B**	**C2k**	**W2k**	**C1k**
My Tiny Model	87.75	87.86	75.87	87.06	84.89	71.66	89.17	77.97
My Large Model	90.01	87.88	77.40	87.98	86.14	73.65	89.81	83.59
ACFD [Zhang et al., 2020a]	90.94	-	-	-	-	-	-	-
Ogawa et al. [Ogawa et al., 2018]	-	76.20	-	79.60	-	-	-	-
Nguyen et al. [Nguyen et al., 2018]	-	-	74.94	-	76.76	-	-	-
Inoue et al. [Inoue et al., 2018]	-	-	-	-	-	70.10	77.30	76.20

Chapter 6

CONCLUSION

In this study, I worked on efficient pre-training for face and body detection models in drawings. First, I introduced a self-supervised teacher-student network to the domain of drawings. I proposed a modified OHEM loss to overcome the false-negative cases caused by the teacher network and equalized the student network's weights to the teacher network's per 500 iterations to prevent distortions in the student network. Without using any labels in my target domain, I outperformed the previous supervised SOTA in DCM 772 [Nguyen et al., 2018] and weakly supervised SOTA [Inoue et al., 2018] in all datasets except Clipart 1k [Inoue et al., 2018], and eBDtheque [Guérin et al., 2013].

By leveraging the existing style transferring methods, I highlighted the importance of using pre-trained weights for this domain adaptation task and the positive effects of using style transfer on the pre-training data. Additionally, I analyzed the individual impacts of the variations and showed that using multiple style transferring variations together provides higher performance.

Secondly, I trained fully supervised models with limited and available labeled data, where the models are initialized with the pre-trained weights. Even with limited drawing data, my model obtained the new SOTA score in most drawing datasets when pre-trained with my pipeline. This finding indicates that efficient pre-training is an important aspect where a low amount of data is available, and a teacher-student network is an effective way of pre-training.

Lastly, I conducted two additional tasks: generation and reconstruction of drawing faces and a simple annotation tool for drawings that can label bounding boxes of panels, faces, bodies, speech bubbles, speech bubble tails, and narrative; identities of the same characters; transcripts of the text boxes; associations of faces, bodies, and speech bubbles. While the annotation tool can be helpful to improve this do-

main in the future, generation and reconstruction studies show that current SOTA StyleGAN models are capable of handling a wide range of styles if enough data is fed to the models. However, glitches and failures of the models still exist and are easy to observe if present.

BIBLIOGRAPHY

[Bai et al., 2022] Bai, Q., Xu, Y., Zhu, J., Xia, W., Yang, Y., and Shen, Y. (2022). High-fidelity gan inversion with padding space.

[Bochkovskiy et al., 2020] Bochkovskiy, A., Wang, C.-Y., and Liao, H.-Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection.

[Brumm et al., 2021] Brumm, A., Oktaviana, A. A., Burhan, B., Hakim, B., Lebe, R., Zhao, J.-x., Sulistyarto, P. H., Ririmasse, M., Adhityatama, S., Sumantri, I., et al. (2021). Oldest cave art found in sulawesi. *Science Advances*, 7(3):eabd4648.

[Cai et al., 2019] Cai, Q., Pan, Y., Ngo, C.-W., Tian, X., Duan, L., and Yao, T. (2019). Exploring object relation in mean teacher for cross-domain detection.

[Chen et al., 2018] Chen, Y., Lai, Y.-K., and Liu, Y.-J. (2018). Cartoongan: Generative adversarial networks for photo cartoonization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9465–9474.

[Deng et al., 2020] Deng, J., Li, W., Chen, Y., and Duan, L. (2020). Unbiased mean teacher for cross-domain object detection.

[Ge et al., 2021] Ge, Z., Liu, S., Wang, F., Li, Z., and Sun, J. (2021). Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*.

[Guérin et al., 2013] Guérin, C., Rigaud, C., Mercier, A., Ammar-Boudjelal, F., Bertet, K., Bouju, A., Burie, J.-C., Louis, G., Ogier, J.-M., and Revel, A. (2013). ebdtheque: a representative database of comics. In *Proceedings of the 12th International Conference on Document Analysis and Recognition (ICDAR)*, pages 1145–1149.

[Hicsonmez et al., 2020] Hicsonmez, S., Samet, N., Akbas, E., and Duygulu, P. (2020). GANILLA: generative adversarial networks for image to illustration translation. *CoRR*, abs/2002.05638.

[Inoue et al., 2018] Inoue, N., Furuta, R., Yamasaki, T., and Aizawa, K. (2018). Cross-domain weakly-supervised object detection through progressive domain adaptation.

[Iyyer et al., 2017] Iyyer, M., Manjunatha, V., Guha, A., Vyas, Y., Boyd-Graber, J., au2, H. D. I., and Davis, L. (2017). The amazing mysteries of the gutter: Drawing inferences between panels in comic book narratives.

[Jiang et al., 2021] Jiang, J., Chen, B., Wang, J., and Long, M. (2021). Decoupled adaptation for cross-domain object detection.

[Karras et al., 2021] Karras, T., Aittala, M., Laine, S., Härkönen, E., Hellsten, J., Lehtinen, J., and Aila, T. (2021). Alias-free generative adversarial networks. In *Proc. NeurIPS*.

[Karras et al., 2018] Karras, T., Laine, S., and Aila, T. (2018). A style-based generator architecture for generative adversarial networks.

[Karras et al., 2020] Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., and Aila, T. (2020). Analyzing and improving the image quality of StyleGAN. In *Proc. CVPR*.

[Lin et al., 2018] Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. (2018). Focal loss for dense object detection.

[Lin et al., 2015] Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., and Dollár, P. (2015). Microsoft coco: Common objects in context.

[Lin et al., 2014] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.

[Liu et al., 2016] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C. (2016). Ssd: Single shot multibox detector. *Lecture Notes in Computer Science*, page 21–37.

[Liu et al., 2021] Liu, Y.-C., Ma, C.-Y., He, Z., Kuo, C.-W., Chen, K., Zhang, P., Wu, B., Kira, Z., and Vajda, P. (2021). Unbiased teacher for semi-supervised object detection.

[Matsui et al., 2017] Matsui, Y., Ito, K., Aramaki, Y., Fujimoto, A., Ogawa, T., Yamasaki, T., and Aizawa, K. (2017). Sketch-based manga retrieval using manga109 dataset. *Multimedia Tools and Applications*, 76(20):21811–21838.

[Nguyen et al., 2018] Nguyen, N.-V., Rigaud, C., and Burie, J.-C. (2018). Digital comics image indexing based on deep learning. *Journal of Imaging*, 4(7).

[Ogawa et al., 2018] Ogawa, T., Otsubo, A., Narita, R., Matsui, Y., Yamasaki, T., and Aizawa, K. (2018). Object detection for comics using manga109 annotations.

[Redmon et al., 2015] Redmon, J., Divvala, S. K., Girshick, R. B., and Farhadi, A. (2015). You only look once: Unified, real-time object detection. *CoRR*, abs/1506.02640.

[Ren et al., 2016] Ren, S., He, K., Girshick, R., and Sun, J. (2016). Faster r-cnn: Towards real-time object detection with region proposal networks.

[Sauer et al., 2022] Sauer, A., Schwarz, K., and Geiger, A. (2022). Stylegan-xl: Scaling stylegan to large diverse datasets.

[Shrivastava et al., 2016] Shrivastava, A., Gupta, A., and Girshick, R. (2016). Training region-based object detectors with online hard example mining.

[Wang and Yu, 2020] Wang, X. and Yu, J. (2020). Learning to cartoonize using white-box cartoon representations. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Xu et al., 2021] Xu, M., Zhang, Z., Hu, H., Wang, J., Wang, L., Wei, F., Bai, X., and Liu, Z. (2021). End-to-end semi-supervised object detection with soft teacher. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

[Yang et al., 2016] Yang, S., Luo, P., Loy, C. C., and Tang, X. (2016). Wider face: A face detection benchmark. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Zhang et al., 2020a] Zhang, B., Li, J., Wang, Y., Cui, Z., Xia, Y., Wang, C., Li, J., and Huang, F. (2020a). Acfd: Asymmetric cartoon face detector.

[Zhang et al., 2020b] Zhang, H., Chang, H., Ma, B., Wang, N., and Chen, X. (2020b). Dynamic R-CNN: Towards high quality object detection via dynamic training. In *ECCV*.

[Zheng et al., 2020] Zheng, Y., Zhao, Y., Ren, M., Yan, H., Lu, X., Liu, J., and Li, J. (2020). Cartoon face recognition: A benchmark dataset. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 2264–2272.

[Zhu et al., 2017] Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*.

[Zhu et al., 2021] Zhu, Y., Cai, H., Zhang, S., Wang, C., and Xiong, Y. (2021). Tinaface: Strong but simple baseline for face detection.

Appendix A

VISUAL RESULTS FROM DETECTORS IN DIFFERENT STAGES

In Figures A.1, A.2, and A.3 you can see our model's visual outputs for sample drawing images from Manga 109, COMICS, and iCartoonFace. These experiments are conducted with a 0.65 confidence threshold and 0.4 NMS threshold for both face and body objects. Overall, moving from stage 1 to stage 2 weights results in a significant increase in detected areas. However, the false positive proposal count also increases with this step. Fine-tuning the model that is initialized with the stage 2 weights successfully suppresses these false positive predictions. By increasing the model size, undetected faces and bodies are further found.

Figure A.1: Sample results from Manga 109 pages. Top-left: stage 1 weights, top-right: stage 2 weights, bottom-left: stage 3 weights, bottom-right: stage 3 XL model weights. Better viewed by zooming.

Figure A.2: Sample results from COMICS pages. Left to right: stage 1 weights, stage 2 weights, stage 3 weights, stage 3 XL model weights. Better viewed by zooming.

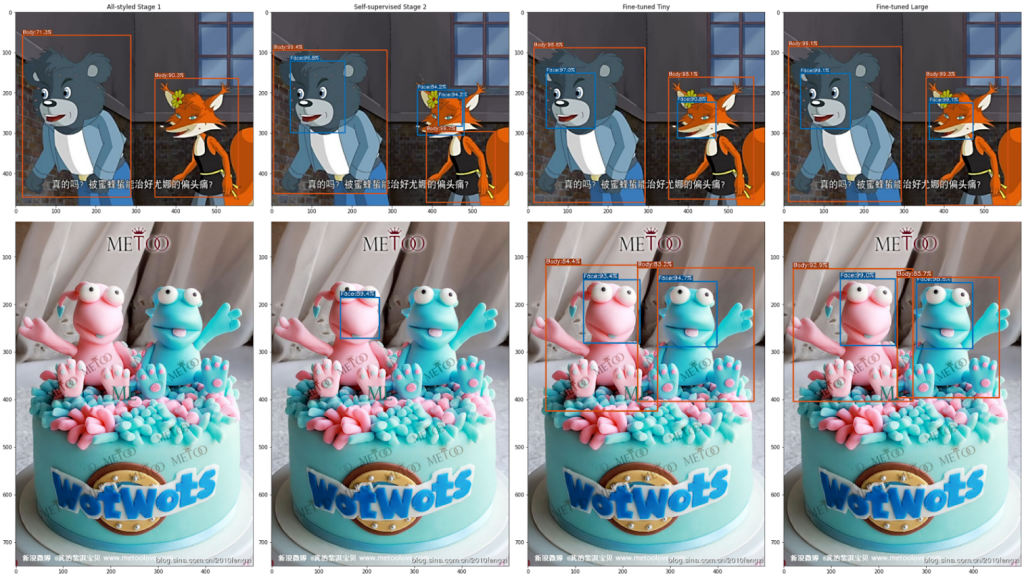

Figure A.3: Sample results from iCartoonFace. Left to right: stage 1 weights, stage 2 weights, stage 3 weights, stage 3 XL model weights. Better viewed by zooming.