Illegible Text to Readable Text: An Image-to-Image Transformation using Conditional Sliced Wasserstein Adversarial Networks

Mostafa Karimi *
Texas A&M University
mostafa_karimi@tamu.edu

Gopalkrishna Veni
Ancestry.com
gveni@ancestry.com

Yen-Yun Yu
Ancestry.com
yyu@ancestry.com

Abstract

Automatic text recognition from ancient handwritten record images is an important problem in the genealogy domain. However, critical challenges such as varying noise conditions, vanishing texts, and variations in handwriting makes the recognition task difficult. We tackle this problem by developing a handwritten-to-machine-print conditional Generative Adversarial network (HW2MP-GAN) model that formulates handwritten recognition as a text-Image-to-text-Image translation problem where a given image, typically in an illegible form, is converted into another image, close to its machine-print form. The proposed model consists of three-components including a generator, and word-level and character-level discriminators. The model incorporates Sliced Wasserstein distance (SWD) and U-Net architectures in HW2MP-GAN for better quality image-to-image transformation. Our experiments reveal that HW2MP-GAN outperforms state-of-the-art baseline cGAN models by almost 30 in Frechet Handwritten Distance (FHD), 0.6 in average Levenshtein distance and 39% in word accuracy for image-to-image translation on IAM database. Further, HW2MP-GAN improves handwritten recognition word accuracy by 1.3% compared to baseline handwritten recognition models on IAM database.

1. Introduction

Text recognition from ancient handwritten record images is an important problem in the genealogy domain helping genealogists discover and unlock family history. Automating the text recognition process would further benefit them in saving time, manual labor and associated cost respectively. However, ancient document images suffer from critical challenges including varying noise conditions, interfering annotations, typical ancient record artifacts like fading and vanishing texts, and variations in handwriting making it difficult to transcribe [27]. Over the past decade, various approaches have been proposed to solve document analysis and recognition such as optical character recognition (OCR) [26], layout analysis [28], text segmentation [19] and handwriting recognition [35, 10, 9, 13]. Although OCR models have been very successful in recognizing machine print text, they stumble upon handwriting recognition due to aforementioned challenges and connecting characters in the text as compared to machine print ones where the characters are easily separable.

Unlike standard techniques that transcribe handwriting images by treating them as either a classification or segmentation problem [42, 21, 43, 5], depending upon the context, we follow a different approach. In essence, we formulate handwriting recognition as a text-Image-to-text-Image translation problem where a given image, typically in an illegible form, is transformed into an another image, closer to machine-print, which can then be easily transcribed using OCR-like techniques. By doing so, high-quality results can be achieved even on extremely challenging handwriting images.

Generative adversarial network (GAN)-based deep generative models have shown a great success in image-to-image translation tasks [6, 12, 25]. Basically, GAN [6] consists of a generator network that tries to map latent space (noise) to the true data distribution while generating fake samples resembling the real ones and a discriminator network that tries to distinguish true samples from the fake ones. Both networks compete against each other until they reach equilibrium. However, GAN inherently suffers from major challenges including non-convergence, mode collapse and a vanishing gradient problem [1]. A variant of GAN called sliced Wasserstein GAN (WGAN) [41] has been introduced to address these challenges. We use a modified version of sliced WGAN into our framework to translate handwritten text images. In the proposed model, we use a U-Net architecture [33] inside the generator as it captures low-level as well as abstract features. For the discriminator part, the proposed model accounts for both word- and character-level errors and underlying high-dimensional distributions leveraged by Wasserstein distance with slice sam-

*Work done during internship at Ancestry.com

pling to transcribe a given text.

The key contributions of the proposed framework include:

- Develop a novel GAN model with three components including one generator and two discriminators. The model generator tries to fool both discriminators to generate realistic "fake" machine print images with realistic characters. The first discriminator, namely a word-level discriminator, tries to distinguish between "fake" machine print generated images and real ones given a handwriting image. The second discriminator is a character level discriminator that tries to distinguish between "fake" character generation and real ones.

- Develop conditional sliced Wasserstein GAN (cSWGAN) model with Lipschitz continuity constraint as a gradient penalty to convert handwritten images into machine print ones.

- Utilize U-Net architecture [33] inside our model generator, similar to the pix2pix model, for good quality image generation.

The rest of the paper is organized as follows. Section 2 reviews the related work. In Section 3, our novel cSWGAN with word- and character-level discriminators is described. Experiments and results are discussed in Section 4. and final conclusions are offered in Section 5.

2. Related works

Handwriting image recognition is traditionally divided into two groups including online [30] and offline recognition [39]. In the online case, the time series of coordinates representing the movement of the pen tip is captured [9] whereas in offline, the image of the text is available. We deal with the latter case. Several computer vision and machine learning algorithms have been proposed to solve various challenges of handwriting recognition [4, 20] but the problem is far from being solved. Some standard handwriting recognition approaches include hidden Markov models (HMM) [38], support vector machines (SVM) [3] and sequential networks including recurrent neural networks (RNN) and its variants.

Sequential networks outperform SVM and HMM models in handwriting recognition tasks which is explained in the following. Long short term memory (LSTM) networks are a type of RNN that propagate sequential information for long periods of time and have been widely applicable in handwriting recognition tasks [9]. Multidimensional Recurrent Neural Networks [8] are another type of sequential networks that have been widely used in modern handwritten text recognition tasks [10]. Annotating handwritten text at a character level is a challenging task. Connectionist Temporal Classification (CTC) [7] has been developed that avoids calculating loss of sequential networks at the character level. Further, CTC-based networks do not require post-processing of the recognized text. Therefore sequential networks with CTC loss has gained a lot of attention in handwriting recognition tasks. The proposed design also uses a similar model as a part of its framework.

As mentioned earlier, Generative adversarial networks (GANs) have proven to be successful generative models in many computer vision tasks. GAN formulates a generative model as a game theory minimax game between generator and discriminator models. The Generator model tries to generate "fake" samples as close to the real ones and the discriminator model tries to discriminate "fake" samples from real ones. An extension of GAN is conditional GAN where the sample generation is conditioned upon an input which can be a discrete label [25], a text [32] or an image [12]. Isola et al., [12] proposed pix2pix GAN that utilizes conditional GAN framework and U-Net architecture [33] for their generator and discriminator models. This approach tends to capture hierarchical features inside images. Although GAN models are very successful in generating fascinating, realistic images [14], they are hard to train due to their difficulty in achieving Nash equilibrium [34], low dimensional support [2], vanishing gradient [23], and mode collapsing [2] issues.

Existing GANs employ either Kullback–Leibler (KL) or Jensen–Shannon (JS) divergence to model loss functions, which could give rise to mode collapsing, gradient vanishing and low dimensional support problems in a high-dimensional space. Wasserstein distance (WD) has gained attention in computer vision and machine learning community due to its everywhere continuous and almost everywhere differentiable nature, which can overcome the above mentioned problems especially the low dimensional support problem. Arjovsky et al., [2] proposed Wasserstein GAN (WGAN), which uses Wasserstein-1 (earth mover) distance to learn probability distributions. The underlying problem with Wasserstein-1 distance is that its primal form is intractable [2] and it is hard to enforce Lipschitz continuity constraint in high-dimensional space for its dual form. To circumvent this problem, sliced Wasserstein Distance (SWD) [41] can be used based on the fact that Wasserstein distance provides a closed-form solution for one-dimensional probability densities. Previously, SWD has been utilized for dimensionality reduction, clustering [18], and learning Gaussian mixture models [17]. Recently, it has been employed in generative models such as sliced Wasserstein generative models [41] and sliced Wasserstein autoencoders [17]. SWD factorizes high-dimensional probabilities to multiple marginal distributions [41]. Theoretically, SWD can compute infinitely many linear projections of a

high-dimensional distribution to one-dimensional distributions followed by computing average Wasserstein distance of these one-dimensional distributions [16].

We have developed a novel conditional sliced Wasserstein GAN with three components including a generator, a word-level discriminator and a character-level discriminator for translating handwritten text images to corresponding machine print forms.

3. Methods

3.1. Generative adversarial networks

GAN can be represented using minimax game framework [6]. Thus, its objective function can be written as:

$$\min_G \max_D \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathbb{P}_r}[\log(D(\mathbf{x}))] + \mathop{\mathbb{E}}_{\tilde{\mathbf{x}}\sim\mathbb{P}_g}[\log(1-D(\tilde{\mathbf{x}}))], \quad (1)$$

where G represents a generator, D represents a discriminator and $\mathbf{x}$ is the realization of true samples. $\mathbb{P}_r$ is the true data distribution and $\mathbb{P}_g$ denotes the generator's distribution that is modeled implicitly by $\tilde{\mathbf{x}} \sim G(\mathbf{z})$ and $\mathbf{z} \sim \mathbb{P}(\mathbf{z})$ (the latent space or noise $\mathbf{z}$ is sampled usually from a uniform distribution or a spherical Gaussian distribution).

Training a GAN network is equivalent to minimizing the Jensen-Shannon (JS) divergence between $\mathbb{P}_r$ and $\mathbb{P}_g$ if the discriminator is trained to optimality before each generator's update [6]. However, it has been observed that Eq. (1) tends to suffer from the gradient vanishing problem as the discriminator saturates. Although generator's loss function can be replaced by maximizing $\mathop{\mathbb{E}}_{\mathbf{z}\sim\mathbb{P}(\mathbf{z})}[\log(D(G(\mathbf{z})))]$, the gradient vanishing problem is far from being solved [6].

Later, GAN has been extended to conditional GAN (cGAN) [25] where both generator and discriminator are conditioned on a given additional supervised event $\mathbf{y}$, where $\mathbf{y}$ can be any kind of auxiliary information or data such as discrete label [25], text [32] and image [12]. Usually cGAN is performed by feeding $\mathbf{y}$ into both discriminator and generator as an additional input layer. cGAN is formulated as:

$$\min_G \max_D \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathbb{P}_r}[\log(D(\mathbf{x}|\mathbf{y}))] + \mathop{\mathbb{E}}_{\tilde{\mathbf{x}}\sim\mathbb{P}_g}[\log(1-D(\tilde{\mathbf{x}}|\mathbf{y}))] \quad (2)$$

where $\mathbb{P}_g$, the generator's distribution, is explicitly modeled as $\tilde{\mathbf{x}} \sim G(\mathbf{z}|\mathbf{y})$ and $\mathbf{z} \sim \mathbb{P}(\mathbf{z})$ in cGAN.

Recently, a variant of GAN in the form of Sliced Wasserstein Generative adversarial network (SWGAN) with gradient penalty constraint [11] has been introduced to stabilize training while generating high-quality image samples. Further, SWGANs provide benefits by tackling convergence and multidimensional intractability issues over traditional GANs [40, 41]. The modified objective function is:

$$\min_G \max_D \int_{\theta\in\mathbb{S}^{n-1}} \Big(\mathop{\mathbb{E}}_{\mathbf{x}\sim\mathbb{P}_r}[D(\mathbf{x})] - \mathop{\mathbb{E}}_{\mathbf{y}\sim\mathbb{P}_g}[D(\mathbf{y})]\, d\theta \Big) + \lambda_1 \mathop{\mathbb{E}}_{\hat{\mathbf{x}}\sim\mathbb{P}_{\hat{X}}}[||\nabla_{\hat{\mathbf{x}}} D(\hat{\mathbf{x}})||_2^2] + \lambda_2 \mathop{\mathbb{E}}_{\hat{\mathbf{y}}\sim\mathbb{P}_{\hat{y}}}[(||\nabla_{\hat{\mathbf{y}}} T(\hat{\mathbf{y}}) - \mathbf{1}||_2^2] \quad (3)$$

where θ represents trainable parameters embedded in D, $\mathbf{1}$ is a vector with all entries equal to 1, λ_1 and λ_2 are the hyper-parameters for balancing the gradient penalty terms and dual SWD. Details on how we reached Eq. (3) from Eq. (2) have been covered in the Supplementary material section.

3.2. Proposed method

In this paper, we have developed a handwritten-to-machine print GAN (HW2MP-GAN) model to pre-process and convert handwritten text images to machine print ones. We consider a *three-component game* between a single generator and two discriminators, which are character- and word-level discriminators, for our conditional GAN model. Two discriminators work together and help the generator in producing clear words and characters in the correct order. The *character*-level discriminator enforces each generated character to be similar to real machine-print characters. Since the number of English characters, symbols and numbers is limited, the *character*-level discriminator's task of learning to generate each one of these characters correctly is easier than the other one. The *word*-level discriminator forces generated words to be similar to the real ones. Since the number of combination of all characters, symbols and numbers is exponential to the length of the word, *word*-level discriminator does a harder task of enforcing the correct order from the generated characters. The overall algorithm is shown in figure 1 (a).

Character level: Suppose, real and generated machine print images are $\mathbf{x}$ and $\tilde{\mathbf{x}}$ respectively. Assume there are $K_{\mathbf{x}}$ characters in the image $\mathbf{x}$. Then, we define the real and generated machine print characters as $\{\mathbf{x}_k^c\}_{k=1}^{K_{\mathbf{x}}}$ and $\{\tilde{\mathbf{x}}_k^c\}_{k=1}^{K_{\mathbf{x}}}$ respectively. Superscript "c" and "w" are used for character-level and word-level respectively. $\mathbf{x}_k^c$ and $\tilde{\mathbf{x}}_k^c$ represent the k^{th} character of word $\mathbf{x}$ and $\tilde{\mathbf{x}}$ respectively. Our character level discriminator is defined as $D^c := \{S_m^c \circ E^c\}_{m=1}^{M^c}$ where E^c is the character level encoder, S_m^c is the m^{th} SWD block and M^c is the number of SWD blocks for character-level discriminator. Therefore, the character-level loss function is formulated as:

$$L^c = \int_{\theta^c\in\mathbb{S}^{r^c-1}} \Big(\mathop{\mathbb{E}}_{\mathbf{x}_k^c\sim\mathbb{P}_r^c}[D^c(\mathbf{x}_k^c)] - \mathop{\mathbb{E}}_{\tilde{\mathbf{x}}_k^c\sim\mathbb{P}_g^c}[D^c(\tilde{\mathbf{x}}_k^c)]\, d\theta \Big) + \lambda_1^c \mathop{\mathbb{E}}_{\hat{\mathbf{x}}_k^c\sim\mathbb{P}_{\hat{\mathbf{x}}}^c}[||\nabla_{\hat{\mathbf{x}}_k^c} D^c(\hat{\mathbf{x}}_k^c)||_2^2] + \lambda_2^c \mathop{\mathbb{E}}_{\bar{\mathbf{x}}_k^c\sim\mathbb{P}_{\bar{\mathbf{x}}}^c}[(||\nabla_{\bar{\mathbf{x}}_k^c} T^c(\bar{\mathbf{x}}_k^c) - \mathbf{1}||_2^2] \quad (4)$$

where the real machine print character distribution is $\mathbb{P}_r^c$ and the generated machine print character distribution

is $\mathbb{P}_g^c$. θ^c represent learnable parameters and are embedded in the character discriminator D^c. The last two terms of Eq. (4) are gradient and Lipschitz regularization terms, where hyper-parameters λ_1^c and λ_2^c are balancing between the SWGAN's loss function and its regularization terms, and $\mathbf{1}$ is the vector of all ones. The gradient and Lipschitz regularization are enforced according to the $\mathbb{P}_{\hat{\mathbf{x}}}^c$ and $\mathbb{P}_{\bar{\mathbf{x}}}^c$ distributions which are sampling across the lines between $\mathbb{P}_r^c$ and $\mathbb{P}_g^c$.

Word level: Similarly to character-level discriminator, our word-level discriminator is defined as $D^w := \{S_m^w \circ E^w\}_{m=1}^{M^w}$ where the E^w is word level encoder, S_m^w is m^{th} SWD block and M^w is the number of SWD blocks. Therefore, the word level loss function is formulated as:

$$L^w = \int_{\theta^w \in \mathbb{S}^{n-1}} \Big(\mathop{\mathbb{E}}_{\mathbf{x}\sim\mathbb{P}_r} [D^w(\mathbf{x}|\mathbf{y})] - \mathop{\mathbb{E}}_{\tilde{\mathbf{x}}\sim\mathbb{P}_g} [D^w(\tilde{\mathbf{x}}|\mathbf{y})] \, d\theta \Big) + \lambda_1^w \mathop{\mathbb{E}}_{\hat{\mathbf{x}}\sim\mathbb{P}_{\hat{x}}} [||\nabla_{\hat{\mathbf{x}}} D^w(\hat{\mathbf{x}}|\mathbf{y})||_2^2] + \lambda_2^w \mathop{\mathbb{E}}_{\bar{\mathbf{x}}\sim\mathbb{P}_{\bar{x}}} [(||\nabla_{\bar{\mathbf{x}}} T^w(\bar{\mathbf{x}}) - \mathbf{1}||_2^2] \Big] \tag{5}$$

where the real machine print word distribution is $\mathbb{P}_r$ and the generated machine print word distribution is $\mathbb{P}_g$. θ^w is the learnable parameters and embedded in the word discriminator D^w. The last two terms are the gradient and Lipschitz regularization terms where hyper-parameters λ_1^w and λ_2^w are balancing between the SWGAN's loss function and its regularization terms. Similarly, the gradient and Lipschitz regularization are enforced according to the $\mathbb{P}_{\hat{\mathbf{x}}}$ and $\mathbb{P}_{\bar{\mathbf{x}}}$ distributions.

Proposed HW2MP-GAN: Our final loss function is combined with character level model, Eq. (4), and word level model, Eq. (5), with reconstruction loss, which is the l_1 norm between generated images $\tilde{x}$ and real images x. L_1 norm has been chosen over l_2 norm since it encourages less bluring both in practice [12] and theory [29] The objective function of HW2MP-GAN is:

$$L^{total} = L^w + \lambda_{char} L^c + \lambda_{recons} \mathop{\mathbb{E}}_{\substack{\mathbf{x}\sim\mathbb{P}_r \\ \tilde{\mathbf{x}}\sim\mathbb{P}_g}} ||\tilde{\mathbf{x}} - \mathbf{x}||_1 \tag{6}$$

where λ_{char} and λ_{recons} are hyper-parameters for balancing between word-level loss, character-level loss and the reconstruction loss functions. To make sure that the projection matrices are orthogonal during training for both character- and word-level discriminators, we follow the Steifel manifold similar to [41].

Our whole pipeline is illustrated in figure 1 (a) and the pseudo-code of our algorithm is written in algorithm (1).

3.3. Handwriting recognition reinforced by HW2MP-GAN

As explained in previous subsections, we have developed a novel conditional GAN model, HW2MP-GAN, for converting handwritten images to machine print ones. We have further developed a novel attention-based handwriting recognition model that exploits both handwritten images and their HW2MP-GAN generated machine print ones for the handwriting recognition task. As a proof-of-concept, we have modified a standard handwriting recognition model developed by Shi et al. [36] to exploit both handwritten and generated machine print images. The baseline model consists of CNN layers followed by bidirectional LSTM layers followed by a Connectionist Temporal Classification (CTC) loss [7]. Further, for posterior decoding of CTC loss to predict the words, we used the recently proposed Word beam search algorithm [35].

We have developed a novel joint attention handwriting

Algorithm 1 HW2MP-GAN

Require: Number of dual SWD blocks for word and character level discriminators are M^w and M^c, batch size b, generator G, word level discriminator $D^w = [S_{d,1}^w \circ E^w, \cdots, S_{d,M^w}^w \circ E^w]^T$ and character level discriminator $D^c = [S_{d,1}^c \circ E^c, \cdots, S_{d,M^c}^c \circ E^c]^T$, latent code dimension for word and character level discriminators are r^w and r^c, Lipschitz constants k^c and k^w, training steps h, training hyper-parameters,etc.

for iter=$1 \cdots n_{max}$ **do**
  for t=$1 \cdots n_{critic}$ **do**
    Sample real data $\{\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\}_{i=1}^m \sim \mathbb{P}_r$
    Sample noise $\{\mathbf{z}^{(i)}\}_{i=1}^m \sim p(\mathbf{z})$
    Sample random number $\{\epsilon_1^{(i)}\}_{i=1}^m, \{\epsilon_2^{(i)}\}_{i=1}^m \sim U[0,1]$
    $\{\tilde{\mathbf{x}}^{(i)}\}_{i=1}^m \leftarrow \{G_\theta(\mathbf{z}^{(i)}|\mathbf{y}^{(i)})\}_{i=1}^m$
    $\{\hat{\mathbf{x}}^{(i)}\}_{i=1}^m \leftarrow \{\epsilon_1^{(i)}\mathbf{x}^{(i)} + (1-\epsilon_1^{(i)})\tilde{\mathbf{x}}^{(i)}\}_{i=1}^m$
    $\{\bar{\mathbf{x}}^{(i)}\}_{i=1}^m \leftarrow \{\epsilon_2^{(i)}\mathbf{x}^{(i)} + (1-\epsilon_2^{(i)})\tilde{\mathbf{x}}^{(i)}\}_{i=1}^m$
    L^w is defined in Eq. (5)
    $\theta^w \leftarrow \text{Adam}(\nabla_{\theta^w} \frac{1}{m} \sum_{i=1}^m L^w, \theta^w, \alpha, \beta_1, \beta_2)$
  end for
  for t=$1 \cdots n_{critic}$ **do**
    Sample real character data $\{\mathbf{x}^{c,(i)}\}_{i=1}^m \sim \mathbb{P}_r$
    Sample noise $\{\mathbf{z}^{(i)}\}_{i=1}^m \sim p(\mathbf{z})$
    Sample random number $\{\epsilon_1^{(i)}\}_{i=1}^m, \{\epsilon_2^{(i)}\}_{i=1}^m \sim U[0,1]$
    $\{\tilde{\mathbf{x}}^{c,(i)}\}_{i=1}^m \leftarrow \{G_\theta(\mathbf{z}^{(i)})\}_{i=1}^m$
    $\{\hat{\mathbf{x}}^{c,(i)}\}_{i=1}^m \leftarrow \{\epsilon_1^{(i)}\mathbf{x}^{c,(i)} + (1-\epsilon_1^{(i)})\tilde{\mathbf{x}}^{c,(i)}\}_{i=1}^m$
    $\{\bar{\mathbf{x}}^{c,(i)}\}_{i=1}^m \leftarrow \{\epsilon_2^{(i)}\mathbf{x}^{c,(i)} + (1-\epsilon_2^{(i)})\tilde{\mathbf{x}}^{c,(i)}\}_{i=1}^m$
    L^c is defined in Eq. (4)
    $\theta^c \leftarrow \text{Adam}(\nabla_{\theta^c} \frac{1}{m} \sum_{i=1}^m L^c, \theta^c, \alpha, \beta_1, \beta_2)$
  end for
  Sample a batch of noises $\{\mathbf{z}^{(i)}\}_{i=1}^m \sim p(\mathbf{z})$
  L^{total} is defined in Eq. (6)
  $\theta^g \leftarrow \text{Adam}(\nabla_{\theta^g} \frac{1}{m} \sum_{i=1}^m L^{total}, \theta^g, \alpha, \beta_1, \beta_2)$
end for

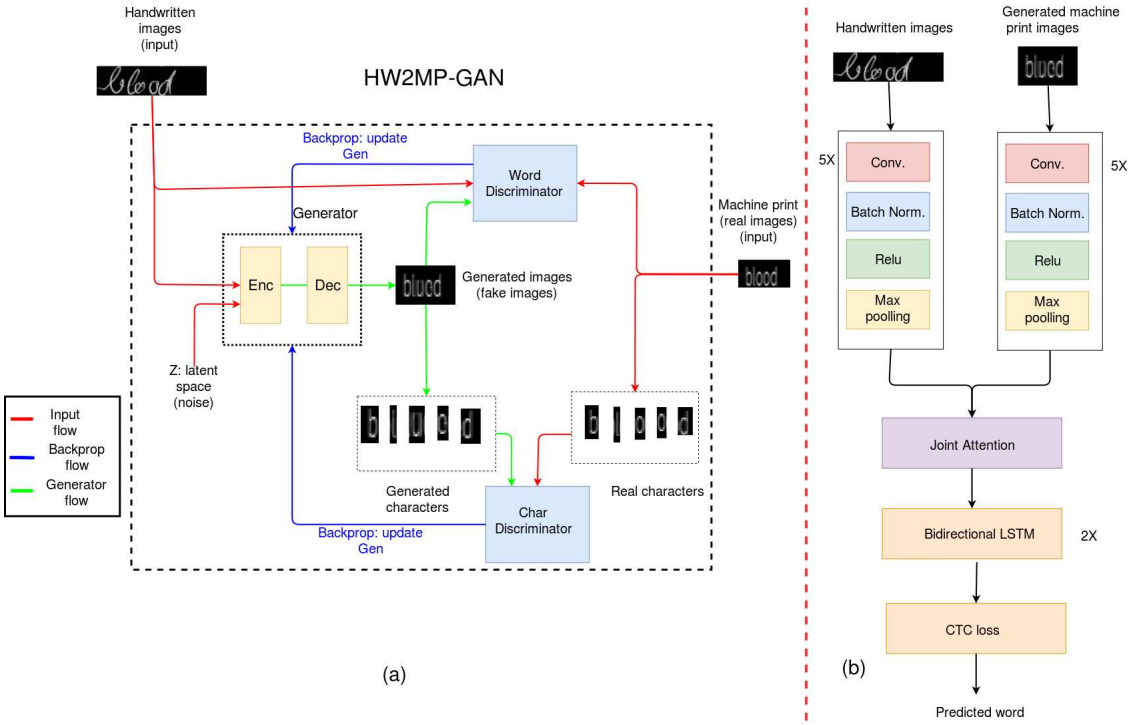

Figure 1: (a) Overall pipeline for HW2MP-GAN (b) Joint attention handwriting recognition reinforced by HW2MP-GAN

recognition model reinforced by HW2MP-GAN as illustrated in Figure 1(b). Our model consists of two parallel series of convolutional layer followed by batch normalization, ReLU nonlinearity and max pooling which is repeated 5 times. These two paths of information have been merged together with a novel joint attention model followed by two layers of Bidirectional LSTMs and CTC loss. The joint attention layer consists of two inputs: 1) features learned from handwritten images denoted by $H = (H_1, \cdots, H_i, \cdots, H_T) \in \mathcal{R}^{T \times d_1}$, and 2) features learned from generated machine print images denoted by $P = (P_1, \cdots, P_j, \cdots, P_T) \in \mathcal{R}^{T \times d_2}$ where T is the maximum length of the word, and d_1 and d_2 represent the number of features for handwritten images and generated machine ones respectively. Therefore, the joint attention layer is formulated as:

$$N_{ij} = tanh(H_i W P_j),\ \alpha_{ij} = \frac{exp(N_{ij})}{\sum_k exp(N_{ik})}\ \forall i, j$$
$$\hat{H}_i = \sum_j \alpha_{ij} P_j \quad \forall i \ ,\ A = Concat(H, \hat{H}) \tag{7}$$

where α_{ij} represents the similarity between the i^{th} handwritten image character and the j^{th} generated machine print character. $\hat{H}_i$ is the projection features learned from the generated machine print image to the handwritten one through attention model. Finally, the output of the attention layer denoted by $A \in \mathcal{R}^{T \times (d_1 + d_2)}$ is a concatenation of the features of handwritten images and their projected ones.

4. Experimental Evaluation

4.1. Data

We evaluated HW2MP-GAN and our joint attention handwriting recognition models on the IAM handwritten database [24]. The IAM database contains 115,320 isolated and labeled words. We randomly chose 95% of the data for our training set and the remaining 5% for our test set. Because IAM images have varying sizes, we resized them to 32×128 pixels. Further, we preprocessed all images by standardizing them to zero-mean and unit-variance.

Training of the HW2MP-GAN model requires handwritten text images and corresponding manually generated machine print forms (i.e., "real" machine print images), which can be created through the ground truth labeled words. Since machine print images contain individual characters, they are used to calculate character-level model loss. Because we have created the "real" machine print images manually, the position of each character is known. Because the number of characters in words varies, we only extracted real or generated characters and ignored the background by enforcing loss zero for the backgrounds.

4.2. Evaluation metrics

For comprehensive evaluation of our model against the state-of-art generative models, we considered three metrics for 1) image-to-image translation problem and 2) handwriting text recognition task. First, Frechet Inception Distance (FID) is the state-of-the-art metric for evaluating the performance of image-to-image generative models. It compares distances between a pair of Inception embedding features from real and generated images [37]. In this paper, we extended the FID score to *Frechet Handwritten Distance* (FHD) to calculate the distance between embedded features of real and model generated text images. The embedded features are computed from the output of bidirectional LSTM layers of the pre-trained handwriting recognition model [1]. Formally, FHD is defined as Frechet distance d(·,·) between the Gaussian $(\mathbf{m}_r, \mathbf{C}_r)$ from the embedded features of real machine print images and the Gaussian $(\mathbf{m}_w, \mathbf{C}_w)$ from the embedded features of GAN generated machine print images which is formulated as:

$$d^2((\mathbf{m}_r, \mathbf{C}_r), (\mathbf{m}_w, \mathbf{C}_w)) = ||\mathbf{m}_r - \mathbf{m}_w||_2^2 + \mathrm{Tr}(\mathbf{C}_r - \mathbf{C}_w + 2(\mathbf{C}_r\mathbf{C}_w)^{1/2}) \quad (8)$$

where Tr is the trace of matrix. FHD=0 is the best and it signifies that the embedded features are identical. For the handwritten text recognition task, this paper used average Levenshtein distance (LD=0 is the best) [22] and word accuracy. Levenshtein distance or "edit distance" is defined as the minimum cost of transforming one string into another through a sequence of weighted edit operations such as insertions, substitutions, and deletions [22].

4.3. Implementation

As explained earlier, our HW2MP-GAN consists of three components: generator, character-level discriminator, and word-level discriminator. Our generator architecture comprise a U-Net model [33] with 5 layers of encoder and decoder each, where encoder and decoder are interconnected through skip connections. The architecture of encoders for both character-level and word-level discriminators are similar to the discriminator in WGAN-GP [11] minus the last linear projection layer. The character-level and word-level encoders embed images to $r^w = 128$ and $r^c = 32$ features respectively. Similar to the original SWGAN [41], we used $M^c = M^w = 4$ SWD blocks for both character-level and word-level discriminator . We chose hyper-parameters based on grid search over a limited set and our results can be further improved by increasing the search space of hyper-parameters. We chose $\lambda_{char} = 2$, $\lambda_{recons} = 100$, $\lambda_1^c = \lambda_1^w = 20$ and $\lambda_2^c = \lambda_2^w = 10$. Adam optimizer [15] with initial learning rate of 0.0001 was used for training the generator and two discriminators.

[1]We pre-trained handwriting recognition model [36] using manually generated machine print images from Sec. 4.1, i.e., built an OCR-like recognition model, whose accuracy $> 99\%$ on machine print text images.

4.4. Text-Image-to-Text-Image translation problem using HW2MP-GAN

This section talks about the performance of the proposed HW2MP-GAN for solving the Text-Image-to-Text-Image translation problem. The experiments include 1) measuring the distance between real machine print images and HW2MP-GAN generated text images, and 2) the legibility of HW2MP-GAN generated text images. To evaluate the legibility, we used a pretrained handwriting recognition model [1] to recognize the HW2MP-GAN generated text images. We compared the HW2MP-GAN model with state-of-the-art GANs that include DCGAN [31], LSGAN [23], WGAN [2], WGAN-GP [11], CTGAN [40], SWGAN [41] and Pix2Pix [12]. In order to put these GANs (except Pix2Pix) in a framework of converting handwriting text images to machine print ones, we further extended them to conditional GAN by embedding handwritten images to a latent space and then concatenating them with noise for machine print generation.

The results of IAM dataset evaluation based on the three metrics including FHD, average LD and word accuracy have been reported in Table 1. Based on our results, we can categorize them into four groups. First, DCGAN and LSGAN models didn't converge due to gradient vanishing problem; Second, WGAN and Pix2Pix models were better than category-1 GAN models since they have improved the GAN model through a better distance metric (Wasserstein in comparison to JS) and better architecture (U-Net model) but have the worst performances compared to other three models. Third, WGAN-GP, CTGAN and SWGAN turned out to be the best baseline models which have comparable results among themselves and outperformed other baseline models. These models as explained, they either have better WD approximation (SWGAN) or better enforcing of Lipschitz continuity constraint (WGAN-GP and CTGAN). Fourth, HW2MP-GAN model outperformed others with a large margin by using all the three metrics. The superior performance of HW2MP-GAN is due to the three-component game, exploiting SWD distance, U-Net architecture and L1 reconstruction loss. However, none of these factors considering alone led to this improvement since for example U-Net architecture and L1 reconstruction loss exist in Pix2Pix model and the SWD distance exists in SWGAN.

Test examples have been illustrated in Figure 2. Based on these results, we can observe that generated machine print images are very similar to the "real" machine print ones. Some errors have been noticed in generating machine print images for example 1) "d" instead of "o" in the word "Almost" 2) "r" instead of "l" in the word "appealed" 3)

Figure 2: Some test examples of converting handwritten images to machine print ones. First row illustrate the handwritten images. Second row shows the generated machine print images and third shows the "real" machine print ones.

"u" instead of "o" in word "without". All of these characters drawn mistakenly are similar to each other which makes it challenging for the generative models.

model	FHD	ave. LD	word accuracy
WGAN	874.76	1.57	0.12%
Pix2Pix	814.24	0.85	5.34%
WGAN-GP	68.57	0.92	16.82%
CTGAN	51.55	0.92	15.48%
SWGAN	60.78	0.94	14.94%
Proposed method	21.42	0.36	55.36%

Table 1: Comparison of GAN models for IAM dataset

4.5. Effect of hidden dimension of LSTM on evaluation metrics

This section talks about evaluating FHD, average LD and word accuracy metrics using different bidirectional LSTM's hidden dimensions in pretrained handwriting recognition models [1]. It also shows that our model consistently outperforms baselines. In Figure 3, hidden dimension $\{16, 32, 64, 128, 256\}$ were used and results showed that 1) HW2MP-GAN, SWGAN, CTGAN and WGAN-GP models maintain consistency in their performance and 2) HW2MP-GAN was superior over all of them for all the hidden dimensions.

4.6. Handwriting recognition reinforced with HW2MP-GAN

We also evaluated the performance of the proposed attention-based handwriting recognition that has been discussed in Section 3.3 on the IAM dataset. The proposed model has been compared against these baselines: handwriting recognition (HWR) models trained by 1) handwritten images alone and 2) generated machine print only. Table 2 shows that the recognition model trained by handwritten text images gains a word accuracy of 84.08% and 0.08 average LS, 62.12% word accuracy and 0.3 average LD by only machine print. Next, the proposed model trained using both results in 85.4% word accuracy and 0.07 average LD. These results demonstrate the potential of exploiting the generated machine print images as an extra source of information to further boost the handwriting recognition task.

model	ave. LD	word accuracy
Handwritten images	0.08	84.08%
Generated machine print images only	0.30	62.12%
Generated machine print + handwritten images	**0.07**	**85.4%**

Table 2: Comparison of HWR models for IAM dataset

5. Conclusion

In this paper, we have demonstrated the advantage of incorporating generative adversarial networks (GANs) in handwriting recognition problems. It has been shown that GAN-based document preprocessing such as handwritten to machine-print image transformation can further improve the accuracy of current handwritten recognition models. Our results on IAM database reveal the superiority of the proposed model on state-of-the-art conditional GAN models for handwritten image to machine-print image translation. Further improvements can be made over the proposed HW2MP-GAN model. Firstly, the model considers image preprocessing and handwritten recognition as separate tasks that can be combined into one. Secondly, current SWD with linear projections can be replaced by generalized SWD with nonlinear projections for more accurate estimate of distances between probabilities.

References

[1] Martin Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks, 2017. 1

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223, 2017. 2, 6

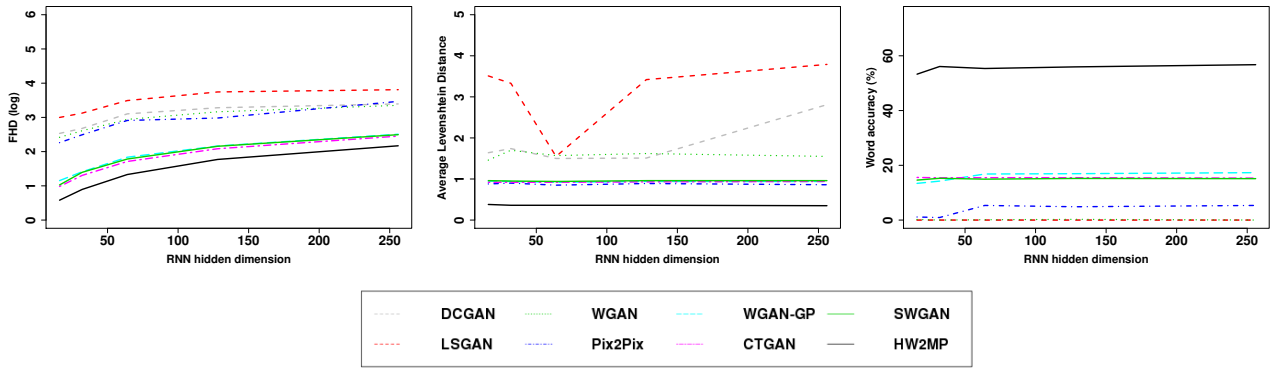

Figure 3: Effect of hidden dimension of bidirectional LSTM in handwriting recognition on the performance of all GAN models in a) FHD b) Average Levenshtein distance c) Word accuracy

[3] Claus Bahlmann, Bernard Haasdonk, and Hans Burkhardt. Online handwriting recognition with support vector machines-a kernel approach. In *Proceedings Eighth International Workshop on Frontiers in Handwriting Recognition*, pages 49–54. IEEE, 2002. 2

[4] Michael Blumenstein, Chun Ki Cheng, and Xin Yu Liu. New preprocessing techniques for handwritten word recognition. In *Proceedings of the second IASTED international conference on visualization, imaging and image processing (VIIP 2002), ACTA Press, Calgary*, pages 480–484, 2002. 2

[5] M-Y Chen, Amlan Kundu, and Sargur N Srihari. Variable duration hidden markov model and morphological segmentation for handwritten word recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 600–601. IEEE, 1993. 1

[6] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 1, 3

[7] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376. ACM, 2006. 2, 4

[8] Alex Graves, Santiago Fernández, and Jürgen Schmidhuber. Multi-dimensional recurrent neural networks. In *International conference on artificial neural networks*, pages 549–558. Springer, 2007. 2

[9] Alex Graves, Marcus Liwicki, Santiago Fernández, Roman Bertolami, Horst Bunke, and Jürgen Schmidhuber. A novel connectionist system for unconstrained handwriting recognition. *IEEE transactions on pattern analysis and machine intelligence*, 31(5):855–868, 2008. 1, 2

[10] Alex Graves and Jürgen Schmidhuber. Offline handwriting recognition with multidimensional recurrent neural networks. In *Advances in neural information processing systems*, pages 545–552, 2009. 1, 2

[11] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in neural information processing systems*, pages 5767–5777, 2017. 3, 6

[12] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017. 1, 2, 3, 4, 6

[13] Junho Jo, Hyung Il Koo, Jae Woong Soh, and Nam Ik Cho. Handwritten text segmentation via end-to-end learning of convolutional neural network. *arXiv preprint arXiv:1906.05229*, 2019. 1

[14] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 2

[15] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[16] Soheil Kolouri, Kimia Nadjahi, Umut Simsekli, Roland Badeau, and Gustavo K Rohde. Generalized sliced wasserstein distances. *arXiv preprint arXiv:1902.00434*, 2019. 3

[17] Soheil Kolouri, Phillip E Pope, Charles E Martin, and Gustavo K Rohde. Sliced-wasserstein autoencoder: an embarrassingly simple generative model. *arXiv preprint arXiv:1804.01947*, 2018. 2

[18] Soheil Kolouri, Yang Zou, and Gustavo K Rohde. Sliced wasserstein kernels for probability distributions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5258–5267, 2016. 2

[19] Omri Koshorek, Adir Cohen, Noam Mor, Michael Rotman, and Jonathan Berant. Text segmentation as a supervised learning task. *arXiv preprint arXiv:1803.09337*, 2018. 1

[20] Gaurav Kumar, Pradeep Kumar Bhatia, and Indu Banger. Analytical review of preprocessing techniques for offline

handwritten character recognition. *International Journal of Advances in Engineering Sciences*, 3(3):14–22, 2013. 2

[21] Yann LeCun, Bernhard E Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne E Hubbard, and Lawrence D Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*, pages 396–404, 1990. 1

[22] Vladimir I Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710, 1966. 6

[23] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2794–2802, 2017. 2, 6

[24] U-V Marti and Horst Bunke. The iam-database: an english sentence database for offline handwriting recognition. *International Journal on Document Analysis and Recognition*, 5(1):39–46, 2002. 5

[25] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 1, 2, 3

[26] Shunji Mori, Hirobumi Nishida, and Hiromitsu Yamada. *Optical character recognition*. John Wiley & Sons, Inc., 1999. 1

[27] M. Murdock, S. Reid, B. Hamilton, and J. Reese. Icdar 2015 competition on text line detection in historical documents. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 1171–1175, 2015. 1

[28] Lawrence O'Gorman. The document spectrum for page layout analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1162–1173, 1993. 1

[29] Xudong Pan, Mi Zhang, and Daizong Ding. Theoretical analysis of image-to-image translation with adversarial learning. *arXiv preprint arXiv:1806.07001*, 2018. 4

[30] Réjean Plamondon and Sargur N Srihari. Online and offline handwriting recognition: a comprehensive survey. *IEEE Transactions on pattern analysis and machine intelligence*, 22(1):63–84, 2000. 2

[31] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 6

[32] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. *arXiv preprint arXiv:1605.05396*, 2016. 2, 3

[33] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 1, 2, 6

[34] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, pages 2234–2242, 2016. 2

[35] Harald Scheidl, Stefan Fiel, and Robert Sablatnig. Word beam search: A connectionist temporal classification decoding algorithm. In *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 253–258. IEEE, 2018. 1, 4

[36] Baoguang Shi, Xiang Bai, and Cong Yao. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE transactions on pattern analysis and machine intelligence*, 39(11):2298–2304, 2016. 4, 6

[37] Konstantin Shmelkov, Cordelia Schmid, and Karteek Alahari. How good is my gan? In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 213–229, 2018. 6

[38] Thad Starner, John Makhoul, Richard Schwartz, and George Chou. On-line cursive handwriting recognition using speech recognition methods. In *Proceedings of ICASSP'94. IEEE International Conference on Acoustics, Speech and Signal Processing*, pages V–125. IEEE, 1994. 2

[39] Charles C. Tappert, Ching Y. Suen, and Toru Wakahara. The state of the art in online handwriting recognition. *IEEE Transactions on pattern analysis and machine intelligence*, 12(8):787–808, 1990. 2

[40] Xiang Wei, Boqing Gong, Zixia Liu, Wei Lu, and Liqiang Wang. Improving the improved training of wasserstein gans: A consistency term and its dual effect. *arXiv preprint arXiv:1803.01541*, 2018. 3, 6

[41] Jiqing Wu, Zhiwu Huang, Dinesh Acharya, Wen Li, Janine Thoma, Danda Pani Paudel, and Luc Van Gool. Sliced wasserstein generative models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3713–3722, 2019. 1, 2, 3, 4, 6

[42] Lei Xu, Adam Krzyzak, and Ching Y Suen. Methods of combining multiple classifiers and their applications to handwriting recognition. *IEEE transactions on systems, man, and cybernetics*, 22(3):418–435, 1992. 1

[43] Qi Zhang, Henry A Rowley, Ahmad A Abdulkader, and Angshuman Guha. Stroke segmentation for template-based cursive handwriting recognition, Nov. 27 2007. US Patent 7,302,099. 1