Modality Conversion of Handwritten Patterns by Cross Variational Autoencoders

Taichi Sumi*, Brian Kenji Iwana*, Hideaki Hayashi*, Seiichi Uchida*
*Advanced Information Technology, Kyushu University, Japan
{brian, hideaki.hayashi, uchida}@human.ait.kyushu-u.ac.jp

***Abstract*—This research attempts to construct a network that can convert online and offline handwritten characters to each other. The proposed network consists of two Variational Auto-Encoders (VAEs) with a shared latent space. The VAEs are trained to generate online and offline handwritten Latin characters simultaneously. In this way, we create a cross-modal VAE (Cross-VAE). During training, the proposed Cross-VAE is trained to minimize the reconstruction loss of the two modalities, the distribution loss of the two VAEs, and a novel third loss called the space sharing loss. This third, space sharing loss is used to encourage the modalities to share the same latent space by calculating the distance between the latent variables. Through the proposed method mutual conversion of online and offline handwritten characters is possible. In this paper, we demonstrate the performance of the Cross-VAE through qualitative and quantitative analysis.**

***Keywords*-variational autoencoder; handwritten character recognition; modality conversion**

I. Introduction

Handwritten characters inherently have two modalities: *image* and *temporal trajectory*. This is because a handwritten character image is comprised of a single or multiple strokes and each stroke is originally generated as a temporal trajectory along with the pen movement. This dual-modality is essential and unique to handwritten characters. Therefore, we can expect unique and more accurate recognition methods and applications by utilizing the dual-modality of handwritten characters. This expectation emphasizes the necessity of the methodologies to convert one modality to the other.

Modality conversion from a temporal trajectory to an image is so-called *inking*. For multi-stroke character recognition, inking is a reasonable strategy to remove stroke-order variations. In the past, many hybrid character recognition methods (e.g., [1]) have been proposed, where two recognition engines are used for the original trajectory pattern and its "inked" image, respectively. In other methods (e.g., [2]), the local direction of the temporal trajectory is embedded into the inked image as an extra feature channel.

Modality conversion from a handwritten character image to a temporal trajectory representing the stroke writing order is so-called *stroke recovery* [3]. Comparing to the inking method, stroke recovery is far more difficult because it is the inverse problem of inferring the lost temporal information from a handwritten image.

In this paper, we propose a Cross-Variational Autoencoder (Cross-VAE), a neural network-based modality conversion method for handwritten characters. Cross-VAE has the ability to convert a handwritten character image into its original temporal trajectory and vice versa. In other words, the Cross-VAE can realize stroke recovery as well as inking by itself. This means that the Cross-VAE can manage the dual-modality of handwritten characters.

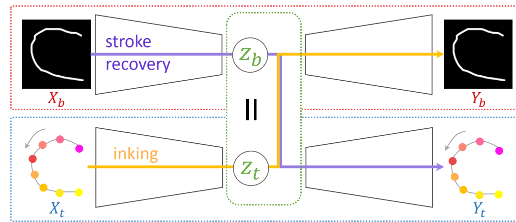


Figure 1. Outline of the proposed Cross-VAE for modality conversion of handwritten characters. Two VAEs are prepared for two modalities, i.e., bitmap image and temporal trajectory, and co-trained so that their latent variables become the same for the same handwritten characters in different modalities. The trained Cross-VAE realizes inking and stroke recovery, as indicated by orange and purple paths, respectively.

As shown in Fig. 1, the Cross-VAE is compounded from two VAEs. Each VAE [4] is a generation model which is decomposed into two neural networks: an encoder that obtains latent variable z from data X and a decoder that obtains output Y close to X from z, i.e., $X \sim Y$. In general, the dimensionality of z is lower than X and Y and thus the latent variable z represents fundamental information of X in a compressed manner. One VAE of Cross-VAE is trained for a handwritten character image (i.e., image $X_\mathrm{b} \to z_\mathrm{b} \to$ image $Y_\mathrm{b}(\sim X_\mathrm{b})$) and the other VAE is trained for a temporal writing trajectory (i.e., temporal trajectory $X_\mathrm{t} \to z_\mathrm{t} \to$ temporal trajectory $Y_\mathrm{t}(\sim X_\mathrm{t})$). Note that the suffixes b and t indicate bitmap image and temporal trajectory, respectively.

The technical highlight of Cross-VAE is that those two VAEs are trained by considering the dual-modality of handwritten characters. Assume that the input image X_b is generated from a temporal trajectory X_t by inking, then we expect that their corresponding latent variables can be the same, that is, $z_\mathrm{b} = z_\mathrm{t}$. This is because X_b and X_t are the same handwritten character in different modalities and thus their fundamental information should be the same. Consequently, if we can co-train two VAEs under the condition $z_\mathrm{b} = z_\mathrm{t}$, we realize, for example, stroke recovery by the following steps: $X_\mathrm{b} \to z_\mathrm{b} = z_\mathrm{t} \to Y_\mathrm{t}(\sim X_\mathrm{t})$.

The main contributions of this paper are summarized as follows:

- A cross-modal VAE is proposed for online and offline handwriting conversion. The Cross-VAE is the combination of two VAEs with different modalities

with a shared latent space and a dual-modality training process.
- A novel loss function called the space sharing loss is introduced. The space sharing loss encourages the latent variables of the VAEs to use the same latent space. The shared latent space is what allows for an input modality to be represented by both output modalities simultaneously.
- Quantitative and qualitative analyses are performed on the proposed method. We show that the Cross-VAE was able to successfully model both online and offline handwriting as well as be used for cross-modal conversion.

II. Related Work

Recently, there are two common approaches that have become popular which use neural networks to learn latent representations, Encoder-Decoders and Generative Adversarial Networks (GAN) [5]. Encoder-Decoders, such as an Autoencoder [6], compress data by encoding the inputs into a latent vector which is then uncompressed by the decoder. The Autoencoder is trained by minimizes the difference between the input and the output of the decoder. GANs take the opposite approach and use a generator, similar to an encoder, then uses a discriminator to maximize the authenticity of the generated data. Where Encoder-Decoders learn the latent representations directly, GANs learn to construct data from random latent representations.

As for cross-modal generation applications, X-Shaped Generative Adversarial Cross-Modal Networks (X-GACMN) [7] creates a shared space for text and images by crossing GANs. Peng et al. [8] also use GANs for text and image entanglement, however, they use weight sharing constraints. Furthermore, a Cross-modal VAE was used by Spurr et al. [9] for hand pose estimation. However, their model only permits multiple pairs of encoders and decoders to share the latent space. Our method trains the VAEs to intertwine with each other and encourages them to share the same latent space. Multi-modal and cross-modal VAEs were also used in [10], [11]. Also, image-to-image translation networks can be seen as a modal conversion. Some examples include CycleGAN [12], StarGAN [13], and UNsupervised Image-to-image Translation (UNIT) [14] networks.

For offline and online handwriting conversion, it has traditionally been done using classical feature-based methods [15] but there has been some recent work using neural networks. Bhunia et al. [16] used a CNN and RNN-based Encoder-Decoder network for handwriting trajectory recovery. Attempts were also made using neural networks to identify graph features [17] and for sequential stroke prediction using regression CNNs [18].

III. Cross-modal Variational Autoencoder (Cross-VAE)

VAEs [4] are Autoencoders which use a variational Bayesian approach to learn the latent representation. VAEs have been used to generate time series data [19], including speech synthesis [20] and language generation [21]. They have also been used for image data [22] and data augmentation [23], [24].

A. Variational Autoencoder (VAE)

A VAE [4] consists of an encoder and a decoder. Given an input $X \in \mathbb{R}^I$, the encoder estimates the posterior distribution of a latent variable $\boldsymbol{z} \in \mathbb{R}^J$.[1] The decoder, in turn, generates an output $Y \in \mathbb{R}^I$ based on a latent variable sampled from the estimated posterior distribution. The VAE is trained end-to-end using a combination of the reconstruction loss $\mathcal{L}_{\mathrm{RE}}$ and the distribution loss $\mathcal{L}_{\mathrm{KL}}$, or:

$$\mathcal{L}_{\mathrm{VAE}} = \mathcal{L}_{\mathrm{KL}} + \mathcal{L}_{\mathrm{RE}}. \tag{1}$$

The reconstruction loss $\mathcal{L}_{\mathrm{RE}}$ is the cross-entropy between the input and the output of the decoder. It is determined by:

$$\mathcal{L}_{\mathrm{RE}} = -\sum_{i=1}^{I} X_i \log Y_i + (1-X_i)\log(1-Y_i), \tag{2}$$

assuming that Y follows the multivariate Bernoulli distribution. In Eq. (2), X_i and Y_i are the i-th element of X and Y, respectively.

The difference between a traditional Autoencoder or Encoder-Decoder network is that the VAE models the latent space using a Gaussian model and uses a variational lower bound to infer the posterior distribution of a latent variable. This is done by including a loss between the latent variables and the unit Gaussian distribution. Specifically, the distribution loss $\mathcal{L}_{\mathrm{KL}}$ is based on the Kullback-Leibler (KL) divergence, or:

$$\mathcal{L}_{\mathrm{KL}} = -\frac{1}{2}\sum_{j=1}^{J}\left(1 + \log(\sigma_j^2) - \mu_j^2 - \sigma_j^2\right), \tag{3}$$

assuming that the prior distribution of the latent variable $\boldsymbol{z}$ follows the multivariate Gaussian distribution of $\mathcal{N}(\mathbf{0}, \mathbf{I})$. In Eq. (3), and μ and σ^2 are the mean and variance of the posterior distribution of $\boldsymbol{z}$.

B. Cross-VAE

We propose the use of a Cross-modal VAE (Cross-VAE) to be used to perform online and offline handwritten character conversion, as illustrated in Fig. 2. The network in red is a VAE for online handwritten characters and the network in blue is for offline handwritten characters. The Cross-VAE is constructed from the joining of two different single

[1] For simplicity, we omit the notation with regard to the number of training data. In the actual calculation, all losses described below are summed over the batch size.

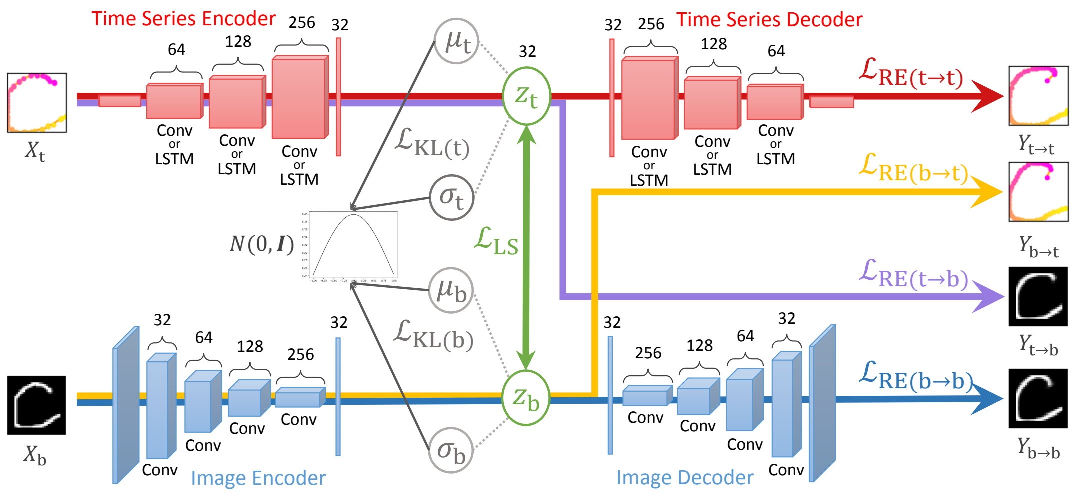


Figure 2. Details of the proposed Cross-VAE. X_{t} is a time series input, X_{b} is an image input. The illustrations of the time series, X_{t}, $Y_{\mathrm{t\to t}}$, and $X_{\mathrm{b\to t}}$, are colored from pink to yellow according to their sequence order. $\mathcal{L}_{\mathrm{KL}}$ is the distribution loss, $\mathcal{L}_{\mathrm{RE}}$ is the reconstruction loss, and $\mathcal{L}_{\mathrm{RE}}$ is the space sharing loss. $Y_{\mathrm{t\to t}}$ and $Y_{\mathrm{b\to b}}$ are the intra-modal outputs and $Y_{\mathrm{t\to b}}$ and $Y_{\mathrm{b\to t}}$ are the cross-modal outputs.

modality VAEs into one multi-modal VAE with a shared cross-modal latent space. Furthermore, we use a cross-modal loss function to ensure that the latent space is shared between the modalities.

During training, the two modalities are trained simultaneously. A time series input X_{t} and an image input X_{b} are entered into the encoders and four outputs are extracted from the decoders. For each input X_{t} and X_{b}, there are respective time series outputs, $Y_{\mathrm{t\to t}}$ and $Y_{\mathrm{b\to t}}$, and respective image outputs $Y_{\mathrm{t\to b}}$ and $Y_{\mathrm{b\to b}}$. The outputs $Y_{\mathrm{t\to t}}$ and $Y_{\mathrm{b\to b}}$ are intra-modal and the outputs $Y_{\mathrm{t\to b}}$ and $Y_{\mathrm{b\to t}}$ are cross-modal.

The loss function of the Cross-VAE is:

$$\mathcal{L}_{\mathrm{Cross}} = \mathcal{L}_{\mathrm{KL}} + \mathcal{L}_{\mathrm{RE}} + \mathcal{L}_{\mathrm{LS}}, \tag{4}$$

where $\mathcal{L}_{\mathrm{KL}}$ is the distribution loss and $\mathcal{L}_{\mathrm{RE}}$ is the reconstruction loss as described in Section III-A. The third loss, $\mathcal{L}_{\mathrm{LS}}$, is the proposed space sharing loss. Due to training with the two inputs, X_{t} and X_{b}, two latent representations are created $\boldsymbol{z}_{\mathrm{t}}$ and $\boldsymbol{z}_{\mathrm{b}}$, respectively. Therefore, the traditional VAE losses, $\mathcal{L}_{\mathrm{KL}}$ and $\mathcal{L}_{\mathrm{RE}}$, need to be modified for Cross-VAE.

Due to the two latent representations, the total distribution loss $\mathcal{L}_{\mathrm{KL}}$ is calculated by combining the individual distribution losses, $\mathcal{L}_{\mathrm{KL(t)}}$ and $\mathcal{L}_{\mathrm{KL(b)}}$, or:

$$\mathcal{L}_{\mathrm{KL}} = \alpha \mathcal{L}_{\mathrm{KL(t)}} + \beta \mathcal{L}_{\mathrm{KL(b)}}, \tag{5}$$

where α and β are weights. The distribution loss of the individual input modalities is calculated using Eq. 3.

Next, the reconstruction loss $\mathcal{L}_{\mathrm{RE}}$ takes into account the reconstruction of $Y_{\mathrm{t\to t}}$ and $Y_{\mathrm{b\to b}}$, as well as the conversion of $Y_{\mathrm{t\to b}}$ and $Y_{\mathrm{b\to t}}$. Thus:

$$\begin{aligned}\mathcal{L}_{\mathrm{RE}} = {} & \gamma_{\mathrm{t\to t}} \mathcal{L}_{\mathrm{RE(t\to t)}} + \gamma_{\mathrm{b\to b}} \mathcal{L}_{\mathrm{RE(b\to b)}} \\ & + \gamma_{\mathrm{t\to b}} \mathcal{L}_{\mathrm{RE(t\to b)}} + \gamma_{\mathrm{b\to t}} \mathcal{L}_{\mathrm{RE(b\to t)}},\end{aligned} \tag{6}$$

where $\mathcal{L}_{\mathrm{RE(t\to t)}}$ and $\mathcal{L}_{\mathrm{RE(b\to t)}}$ are the losses calculated by Eq. (2) to input X_{t} and $\mathcal{L}_{\mathrm{RE(b\to b)}}$ and $\mathcal{L}_{\mathrm{RE(t\to b)}}$ are to input X_{b}. Also, $\gamma_{\mathrm{t\to t}}$, $\gamma_{\mathrm{b\to b}}$, $\gamma_{\mathrm{t\to b}}$, $\gamma_{\mathrm{b\to t}}$ are weight of each respective loss.

(a) Online Handwriting (b) Offline Handwriting

Figure 3. Examples of images created from time series in the experiments. In (a), pink indicates the beginning of the sequence.

C. Space Sharing Loss

While the Cross-VAE is trained using the combination of the reconstruction and distribution losses for the different modalities, we propose the use of a space sharing loss function to encourage the latent variable to share the same latent space. The space sharing loss $\mathcal{L}_{\mathrm{LS}}$ gives the square error of the latent variable $\boldsymbol{z}_{\mathrm{t}}$ obtained from the online character VAE and the latent variable $\boldsymbol{z}_{\mathrm{b}}$ of the offline character VAE. Specifically:

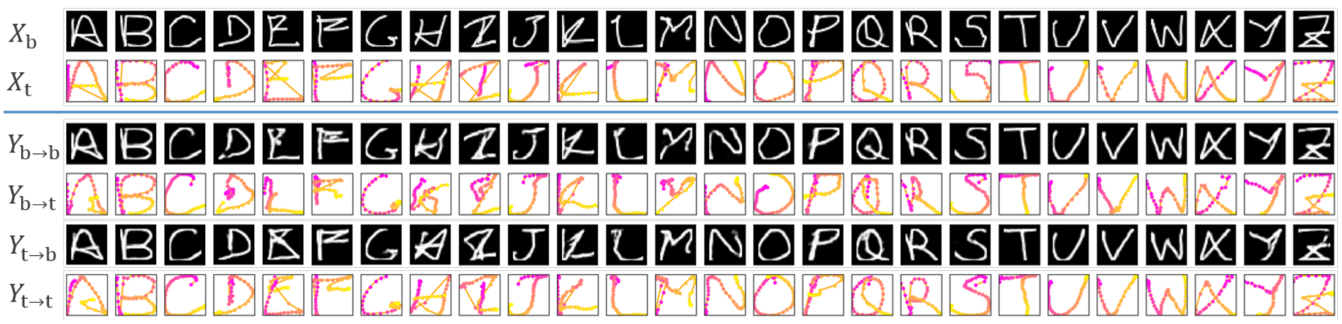
$$\mathcal{L}_{\mathrm{LS}} = \delta \frac{1}{2} \|\boldsymbol{z}_{\mathrm{t}} - \boldsymbol{z}_{\mathrm{b}}\|^2, \tag{7}$$

where δ is a weight and $\|\cdot\|$ is the Euclidean norm.

IV. Online and Offline Conversion of Handwritten Characters using Cross-VAE

A. Dataset

For the experiment, we used handwritten uppercase characters from the Unipen online handwritten character dataset [25]. The online handwritten characters consist of time series made of (x, y) coordinates. The online characters were normalized to fit within a square bound by $(0, 0)$ and $(1, 1)$. In order to use a second modality, the online characters were rendered into images. The images were



(a) Result using LSTM layers for the online encoder and decoder

(b) Result using convolutional layers for the online encoder and decoder

Figure 4. Result of the Cross-VAE. X_b is the original image and X_t is the original time series. $Y_{\mathrm{b}\to\mathrm{b}}$ and $Y_{\mathrm{t}\to\mathrm{t}}$ are outputs of the Cross-VAE which correspond to the same modalities and $Y_{\mathrm{b}\to\mathrm{t}}$ and $Y_{\mathrm{t}\to\mathrm{b}}$ are between different modalities. The illustrations of the time series, X_t, $Y_{\mathrm{t}\to\mathrm{t}}$, and $X_{\mathrm{b}\to\mathrm{t}}$ are colored from pink to yellow according to their sequence order.

32×32 pixels with 0 as the background and 1 as the foreground. Examples of the image renderings can be found in Fig. 3.

B. Architecture Details

The image-based encoder and decoder were constructed from a Convolutional Neural Network (CNN) with a similar structure as a ConvDeconv network [26]. The image encoder consists of four 3×3 convolutional layers with Rectified Linear Unit (ReLU) activations and corresponding 2×2 maxpooling layers. The number of nodes are detailed in Fig. 2. The decoder is a reflection of the encoder which uses unpooling and deconvolutions. Between the convolutional layers, there exist three fully-connected layers. One belonging to each, the encoder and decoder, and one for the latent variable.

For the time series-based encoder and decoder, there were two architectures chosen. The first is a CNN-based approach with 1D convolutions and no pooling. The second is a Recurrent Neural Network (RNN) approach using Long Short Term Memory (LSTM) [27] layers. Both the CNN-based approach and the LSTM-based approach have three fully-connected layers, one for the encoder, one for the latent variable, and one for the decoder. The two layer types were chosen to compare the difference between the LSTM layers which were designed specifically for time series and convolutional layers which are traditionally used for images.

The Cross-VAE was optimized with RMSProp [28] for 200 epochs. The weighting factors of each loss function were determined through experiments. Specifically, they are $\alpha = 0.5$, $\beta = 0.5$, $\gamma_{\mathrm{t}\to\mathrm{t}} = 0.4$, $\gamma_{\mathrm{b}\to\mathrm{b}} = 0.5$, $\gamma_{\mathrm{t}\to\mathrm{b}} = 0.4$, $\gamma_{\mathrm{b}\to\mathrm{t}} = 0.2$, $\delta = 1.0$. The number of dimensions of the latent variable was 32 in all experiments.

C. Conversion Result

The results of the Cross-VAE are shown in Fig. 4. Fig. 4 (a) is from using LSTM layers for the online encoder and decoder and Fig. 4 (b) is from using convolutional layers in the online encoder and decoder. The results $Y_{\mathrm{b}\to\mathrm{b}}$ and $Y_{\mathrm{t}\to\mathrm{b}}$ are the images generated by the inputs X_t and X_b, respectively. The results $Y_{\mathrm{t}\to\mathrm{t}}$ and $Y_{\mathrm{b}\to\mathrm{t}}$ are renderings of the time series colored from pink to yellow in chronological order. Notably, the output $Y_{\mathrm{b}\to\mathrm{t}}$ is the trajectory prediction based on the image input X_b.

By examining Fig. 4, it can be seen that the mutual conversion of the modalities was accurately performed. This shows that the shared latent space learned by the simultaneous encoding of X_b and X_t is able to accurately represent both image data and time series data. In addition, not only was the stroke trajectory inferred, the results show that the shared latent space was able to encode temporal information about what is expected from the characters. For example, the “B” in Fig. 4 (a) is missing information, yet the time series results $Y_{\mathrm{b}\to\mathrm{t}}$ and $Y_{\mathrm{t}\to\mathrm{t}}$ were able to restore the character. The results from Fig. 4 qualitatively confirm

Figure 5. Multiple example results for the letter "A" using convolutional layers for the online encoder and decoder

that the Cross-VAE is able to do mutual modality conversion between the online and offline handwritten characters.

The letter "A" is another character that would normally be difficult to recover lost time series information due to having multiple variations. In some cases, the left-most stroke is drawn downwards and in some, it is drawn upwards depending on the author. Fig. 5 are examples of many different "A"s generated by the Cross-VAE. The figure shows that the Cross-VAE was able to correctly estimate most of the strokes of the "A"s. In particular, the results from $Y_{b\to t}$ was able to not only correctly predict the stroke order but also was able to replicate the stroke velocity. Note the stroke that crosses the center of the "A." This further enforces the success of the proposed Cross-VAE.

D. Quantitative Evaluation of Conversion

In order to evaluate the method quantitatively, we constructed the following three measures to determine the quality of the generated characters:

PSNR: Peak signal-to-noise ratio (PSNR) calculates the similarity between the input images and the generated output images. PSNR is the ratio between the maximum luminance MAX and the amount of noise, or:

$$\mathrm{PSNR} = 10\log_{10}\frac{\mathrm{MAX}^2}{\mathrm{MSE}}, \quad (8)$$

where MSE is the mean squared error between X_b and $Y_{t\to b}$. PSNR is measured in decibels (dB) with a larger value being better.

SSIM: Structural Similarity (SSIM) predicts the perceived difference between images. Similar to PSNR, this acts as a similarity measure between X_b and $Y_{t\to b}$. The equation for SSIM is:

$$\mathrm{SSIM} = \frac{(2\mu_{X_b}\mu_{Y_{t\to b}} + C_1) + (2\sigma_{X_b Y_{t\to b}} + C_2)}{(\mu^2_{X_b} + \mu^2_{Y_{t\to b}} + C_1)(\sigma^2_{X_b} + \sigma^2_{Y_{t\to b}} + C_2)}, \quad (9)$$

where C_1 and C_2 are stabilizing constants set to $C_1 = (0.01 \times 255)^2$ and $C_2 = (0.03 \times 255)^2$. μ is the average luminance, σ^2 is the variance, and σ is the covariance. SSIM is a value from 0 to 1 with a larger value meaning more similar.

Table I
CROSS-CONVERSION EVALUATIONS

	$Y_{\mathrm{t}\to\mathrm{b}}$		$Y_{\mathrm{b}\to\mathrm{t}}$
	PSNR	SSIM	DTW
Cross-VAE (LSTM)	15.26	0.617	0.0411
Cross-VAE (Conv)	15.99	0.707	0.0361
Class Average	9.197	0.159	0.206

DTW: Dynamic time warping (DTW) was used as an evaluation for the time series generation as a method of measuring the stroke trajectory estimation. DTW is a robust distance measure between time series which uses dynamic programming to optimally match sequence elements. In this case, we use the average DTW-distance between the input time series X_{t} and the cross-modality output $X_{\mathrm{t}\to\mathrm{b}}$. Smaller the DTW-distances between X_{t} and $X_{\mathrm{t}\to\mathrm{b}}$ means that the patterns are more similar and the Cross-VAE was able to replicate the original input time series. Thus, a smaller value is better.

The results of quantitative evaluations are shown in Table I. In the table, we evaluate the difference between using LSTM layers and convolutional layers in the time series encoder and decoder. The results are compared to the images and time series of the average pattern in each respective class. PSNR and SSIM are used for the cross-modal conversion from X_{t} to $Y_{\mathrm{t}\to\mathrm{b}}$ and DTW is used for the evaluation of the cross-modal conversion from X_{b} to $Y_{\mathrm{b}\to\mathrm{t}}$.

For online to offline handwritten character conversion, or inking, the Cross-VAE did much better than the class average. In addition, the time series encoder and decoder with convolutional layers performed better than the LSTM. This shows that, despite being time series data, the convolutional layers were able to encode the information into the latent space better than the LSTM layers.

Similarly, for the offline to online handwritten character conversion, the Cross-VAE performed better than the average and the convolutional layer based time series encoder and decoder did better in reconstructing the time series. The DTW results specifically demonstrate that the Cross-VAE is able to predict the trajectories of the strokes. This information is normally lost during the rendering, however, the Cross-VAE is able to infer the stroke trajectory from the shared latent space.

Both evaluations found that using convolutional layers was better than using LSTM layers. This is justified for this data target because handwritten characters are spatial coordinates where the relevance of every element depends on its neighbors. Structured data such as this is well suited to convolutional layers, whereas the advantages of maintaining long-term dependencies in LSTMs is lost. We believe that due to this, the convolutional layer based encoder and decoder for the time series modality produces better results.

V. Conclusion

In this paper, we proposed a VAE for mutual modality conversion called a Cross-VAE. The Cross-VAE is made from the merging of two VAEs of different modalities by enforcing a shared latent space. To train the Cross-VAE, we propose using the combination of reconstruction loss and distribution loss from the original VAE and an additional space sharing loss. The space sharing loss encourages the different modalities of the Cross-VAE to use the same latent space embedding. In the experiments, we used online and offline handwritten characters to verify the ability of the Cross-VAE. The results show that the mutual conversion was possible and that the proposed Cross-VAE could accurately reconstruct the images and time series.

In the future, we will continue to improve the model and apply it to other applications. The Cross-VAE can be used for other types of data and tackle other tasks. Furthermore, this work opens the way for embedding different modalities into one shared latent space which can be used as a tool for representing those modalities in one space.

Acknowledgement

This work was supported by JSPS KAKENHI Grant Number JP17H06100.

References

[1] H. Tanaka, K. Nakajima, K. Ishigaki, K. Akiyama, and M. Nakagawa, "Hybrid pen-input character recognition system based on integration of online-offline recognition," in *IAPR Int. Conf. Document Analysis and Recognition*, 1999, pp. 209–212.

[2] M. Hamanaka, K. Yamada, and J. Tsukumo, "On-line japanese character recognition experiments by an off-line method based on normalization-cooperated feature extraction," in *IAPR Int. Conf. Document Analysis and Recognition*, 1993, pp. 204–207.

[3] V. Nguyen and M. Blumenstein, "Techniques for static handwriting trajectory recovery: A survey," in *IAPR Int. Workshop on Document Analysis Systems*, 2010.

[4] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *Int. Conf. Learning Representations*, 2013.

[5] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, 2014, pp. 2672–2680.

[6] P. Baldi, "Autoencoders, unsupervised learning, and deep architectures," in *ICML Workshop Unsupervised and Transfer Learning*, 2012, pp. 37–49.

[7] W. Guo, J. Liang, X. Kong, L. Song, and R. He, "X-gacmn: An x-shaped generative adversarial cross-modal network with hypersphere embedding," in *Asian Conf. Computer Vision*, 2018.

[8] Y. Peng and J. Qi, "CM-GANs," *ACM Trans. Multimedia Computing, Communications, and Applications*, vol. 15, no. 1, pp. 1–24, feb 2019.

[9] A. Spurr, J. Song, S. Park, and O. Hilliges, "Cross-modal deep variational hand pose estimation," in *IEEE Conf. Computer Vision and Pattern Recognition*, 2018.

[10] F. Huang, X. Zhang, C. Li, Z. Li, Y. He, and Z. Zhao, "Multimodal network embedding via attention based multi-view variational autoencoder," in *ACM Int. Conf. Multimedia Retrieval*, 2018.

[11] I. V. Serban, A. G. Ororbia II, J. Pineau, and A. Courville, "Multi-modal variational encoder-decoders," in *Int. Conf. Learning Representations*, 2016.

[12] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *IEEE Int. Conf. on Computer Vision*, 2017.

[13] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2018.

[14] M.-Y. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," in *Advances in Neural Information Processing Systems*, 2017, pp. 700–708.

[15] V. Nguyen and M. Blumenstein, "Techniques for static handwriting trajectory recovery," in *IAPR Int. Workshop Document Analysis Systems*. ACM Press, 2010.

[16] A. K. Bhunia, A. Bhowmick, A. K. Bhunia, A. Konwer, P. Banerjee, P. P. Roy, and U. Pal, "Handwriting trajectory recovery using end-to-end deep encoder-decoder network," in *Int. Conf. Pattern Recognition*, 2018.

[17] Y. Qiao, M. Nishiara, and M. Yasuhara, "A framework toward restoration of writing order from single-stroked handwriting image," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 28, no. 11, pp. 1724–1737, nov 2006.

[18] B. Zhao, M. Yang, and J. Tao, "Pen tip motion prediction for handwriting drawing order recovery using deep neural network," in *Int. Conf. Pattern Recognition*, 2018.

[19] O. Fabius and J. R. van Amersfoort, "Variational recurrent auto-encoders," in *ICLR Workshop*, 2014.

[20] K. Akuzawa, Y. Iwasawa, and Y. Matsuo, "Expressive speech synthesis via modeling expressions with variational autoencoder," in *Interspeech*, 2018.

[21] S. R. Bowman, L. Vilnis, O. Vinyals, A. Dai, R. Jozefowicz, and S. Bengio, "Generating sentences from a continuous space," in *SIGNLL Conf. Computational Natural Language Learning*. Association for Computational Linguistics, 2016.

[22] Y. Pu, Z. Gan, R. Henao, X. Yuan, C. Li, A. Stevens, and L. Carin, "Variational autoencoder for deep learning of images, labels and captions," in *Advances in Neural Information Processing Systems*, 2016, pp. 2352–2360.

[23] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling, "Semi-supervised learning with deep generative models," in *Advances in Neural Information Processing Systems*, 2014, pp. 3581–3589.

[24] Z. Wan, Y. Zhang, and H. He, "Variational autoencoder based synthetic data generation for imbalanced learning," in *IEEE Symposium Series Computational Intelligence*, 2017, pp. 1–7.

[25] I. Guyon, L. Schomaker, R. Plamondon, M. Liberman, and S. Janet, "UNIPEN project of on-line data exchange and recognizer benchmarks," in *Int. Conf. on Pattern Recognition*, vol. 2, 1994, pp. 29–33.

[26] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *IEEE Int. Conf. Computer Vision*, 2015.

[27] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[28] G. Hinton, N. Srivastava, and K. Swersky, "Neural networks for machine learning lecture 6a overview of mini-batch gradient descent."