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.

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 $X_b \rightarrow z_b \rightarrow$ image $Y_b(\sim X_b)$) and the other VAE is trained for a temporal writing trajectory (i.e., temporal trajectory $X_t \rightarrow z_t \rightarrow$ temporal trajectory $Y_t(\sim X_t)$). 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, $z_b = z_t$. 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 $z_b = z_t$, we realize, for example, stroke recovery by the following steps: $X_b \rightarrow z_b = z_t \rightarrow Y_t(\sim X_t)$.

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.
II. RELATED WORK

Recently, there are two common approaches that have become popular which use neural networks to learn latent representations, Encoder-Decoder and Generative Adversarial Networks (GAN) [5]. Encoder-Decoder, 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 minimizing 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-Decoder 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 cross 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, 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 $z \in \mathbb{R}^J$. 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 $L_{RE}$ and the distribution loss $L_{KL}$, or:

$$L_{VAE} = L_{KL} + L_{RE}.$$  (1)

The reconstruction loss $L_{RE}$ is the cross-entropy between the input and the output of the decoder. It is determined by:

$$L_{RE} = -\sum_{i=1}^{I} X_i \log Y_i + (1-X_i) \log (1-Y_i),$$  (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 $L_{KL}$ is based on the Kullback-Leibler (KL) divergence, or:

$$L_{KL} = -\frac{1}{2} \sum_{j=1}^{J} (1 + \log (\sigma_j^2) - \mu_j^2 - \sigma_j^2),$$  (3)

assuming that the prior distribution of the latent variable $z$ follows the multivariate Gaussian distribution of $N(0, I)$. In Eq. (3), $\mu$ and $\sigma^2$ are the mean and variance of the posterior distribution of $z$.

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

\footnote{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.}
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, $Y_{t\rightarrow t}$ and $Y_{b\rightarrow t}$, and respective image outputs $Y_{t\rightarrow b}$ and $Y_{b\rightarrow b}$. The outputs $Y_{t\rightarrow t}$ and $Y_{b\rightarrow b}$ are intra-modal and the outputs $Y_{t\rightarrow b}$ and $Y_{b\rightarrow t}$ are cross-modal.

The loss function of the Cross-VAE is:

$$\mathcal{L}_{Cross} = \mathcal{L}_{KL} + \mathcal{L}_{RE} + \mathcal{L}_{LS},$$  \hspace{1cm} (4)

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

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

$$\mathcal{L}_{KL} = \alpha \mathcal{L}_{KL(t)} + \beta \mathcal{L}_{KL(b)},$$  \hspace{1cm} (5)

where $\alpha$ and $\beta$ are weights. The distribution loss of the individual input modalities is calculated using Eq. 3.

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

$$\mathcal{L}_{RE} = \gamma_{t\rightarrow t} \mathcal{L}_{RE(t\rightarrow t)} + \gamma_{b\rightarrow b} \mathcal{L}_{RE(b\rightarrow b)} + \gamma_{t\rightarrow b} \mathcal{L}_{RE(t\rightarrow b)} + \gamma_{b\rightarrow t} \mathcal{L}_{RE(b\rightarrow t)},$$  \hspace{1cm} (6)

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

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 $\mathcal{L}_{LS}$ gives the square error of the latent variable $z_t$ obtained from the online character VAE and the latent variable $z_b$ of the offline character VAE. Specifically:

$$\mathcal{L}_{LS} = \delta \frac{1}{2} \| z_t - z_b \|^2,$$  \hspace{1cm} (7)

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

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

C. Conversion Result

The results of the Cross-V AE 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 $Y_{b\rightarrow b}$ and $Y_{t\rightarrow t}$ are the images generated by the inputs $X_t$ and $X_b$, respectively. The results $Y_{t\rightarrow t}$ and $Y_{b\rightarrow t}$ are renderings of the time series colored from pink to yellow in chronological order. Notably, the output $Y_{b\rightarrow t}$ 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 $Y_{b\rightarrow t}$ and $Y_{t\rightarrow t}$ were able to restore the character. The results from Fig. 4 qualitatively confirm
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 is 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 $Y_{b\rightarrow t}$ 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 $\text{MAX}$ and the amount of noise, or:

$$\text{PSNR} = 10 \log_{10} \frac{\text{MAX}^2}{\text{MSE}},$$  \hspace{1cm} (8)

where MSE is the mean squared error between $X_b$ and $Y_{t\rightarrow b}$. 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 $Y_{t\rightarrow b}$. The equation for SSIM is:

$$\text{SSIM} = \frac{(2\mu_X\mu_Y + C_1)(2\sigma_{XY} + C_2)}{(\mu_X^2 + \mu_Y^2 + C_1)(\sigma_X^2 + \sigma_Y^2 + C_2)},$$  \hspace{1cm} (9)

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

**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 $X_{t\rightarrow b}$. Smaller the DTW-distances between $X_t$ and $X_{t\rightarrow b}$ 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 $Y_{t\rightarrow b}$, and DTW is used for the evaluation of the cross-modal conversion from $X_b$ to $Y_{b\rightarrow t}$.

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.

**Table I**

<table>
<thead>
<tr>
<th>Cross-Conversion Evaluations</th>
<th>$Y_{t\rightarrow b}$</th>
<th>$Y_{b\rightarrow t}$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PSNR</strong></td>
<td>Cross-VAE (LSTM)</td>
<td>15.26</td>
</tr>
<tr>
<td></td>
<td>Cross-VAE (Conv)</td>
<td>15.99</td>
</tr>
<tr>
<td><strong>SSIM</strong></td>
<td>Class Average</td>
<td>9.197</td>
</tr>
<tr>
<td><strong>DTW</strong></td>
<td>Cross-VAE (LSTM)</td>
<td>0.0411</td>
</tr>
<tr>
<td></td>
<td>Cross-VAE (Conv)</td>
<td>0.0361</td>
</tr>
<tr>
<td></td>
<td>Class Average</td>
<td>0.206</td>
</tr>
</tbody>
</table>

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

![Figure 5](image-url)
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


