Shared Latent Space of Font Shapes and Their Noisy Impressions

Jihun Kang¹, Daichi Haraguchi¹, Seiya Matsuda¹, Akisato Kimura², and Seiichi Uchida¹

¹ Kyushu University, Fukuoka, Japan {jihun.kang}@human.ait.kyushu-u.ac.jp
² NTT Communication Science Laboratories, NTT Corporation, Japan

Abstract. Styles of typefaces or fonts are often associated with specific impressions, such as heavy, contemporary, or elegant. This indicates that there are certain correlations between font shapes and their impressions. To understand the correlations, this paper realizes a shared latent space where a font and its impressions are embedded nearby. The difficulty is that the impression words attached to a font are often very noisy. This is because impression words are very subjective and diverse. More importantly, some impression words have no direct relevance to the font shapes and will disturb the realization of the shared latent space. We, therefore, use DeepSets for enhancing shape-relevant words and suppressing shape irrelevant words automatically while training the shared latent space. Quantitative and qualitative experimental results with a large-scale font-impression dataset demonstrate that the shared latent space by the proposed method describes the correlation appropriately, especially for the shape-relevant impression words.

Keywords: Font shape · Font impression · Shared latent space.

1 Introduction

Font is multi-modal. This is because a font is comprised of not only a set of visible letter shapes (from ‘A’ to ‘z’) but only a set of impressions. Fig. 1 shows several examples of fonts and their impressions from MyFonts dataset [1]. For example, the font 4-square is tagged with a set of impression words {heavy, headline, display, logo, square}. This is an interesting phenomenon in that a shape gives a specific impression; however, the correlation between shape and impression is not well studied in a reliable and objective data-driven analysis.

Our research aims to realize a shared latent space of the two modalities in order to understand their correlation. Fig. 2 illustrates the shared latent space. Let \( X_i \) denote the \( i \)-th font (i.e., a set of images from ‘A’ to ‘Z’ of the \( i \)-th font) and assume a set of \( J_i \) impression words \( W_i = \{w_{i,1}, \ldots, w_{i,J_i}, \ldots, w_{i,J_i}\} \) are attached to the font \( X_i \). In the \( d \)-dimensional shared latent space, we expect that \( f(X_i) \sim g(W_i) \) for all \( i \), where the embedding functions \( f \) and \( g \) give \( d \)-dimensional representations of \( X_i \) and \( W_i \), respectively. Therefore, the realization of the latent space is the task of getting the representation functions \( f \) and \( g \) that satisfy this proximity condition.
For realizing the shared latent space, we need to deal with the noisiness of the impression words. The noisiness comes from two reasons. First, the impression of a font is subjective and will be variable with its observers. The second and more serious reason is that impression words are often irrelevant to font shape. In Fig. 1 an impression soccer is attached to the font international. This impression might be attached because the font is used for the uniform of a soccer team. As revealed by this example, there are two types of impression words, shape-relevant and shape-irrelevant. The former type (such as sans-serif and heavy) is our target; however, the latter will disturb the realization of the shared latent space and its effect should be weakened.

This paper proposes a novel method for realizing the shared latent space while weakening the effect of noisy (i.e., shape-irrelevant) impression words. Fig. 3 shows the overall structure of the proposed method. It is a cross-modal embedding scheme and comprised of two autoencoders for word and image modalities. These autoencoders are co-trained so that $f(X_i) \sim g(W_i)$, while guaranteeing accurate reconstruction at each modality. Once we realize the shared latent space, it can be used for several applications, such as font image retrieval and
font image generation, as shown in Fig. 3, given a set of impression words, we can retrieve several existing font images or generate new font images.

A technical highlight of the proposed method is that it employs DeepSets [2] for weakening shape-irrelevant impression words. Roughly speaking, DeepSets accepts a set as its input, converts each element into a feature vector internally, and finally outputs the average of feature vectors. If an element of the set is useless for a task, its feature vector will become close to a zero vector and thus its effect on the final output is minimized. In our case, this set corresponds to a set of impression words, and the effect of the impression word that disturbs our task will be minimized. Note that DeepSets is also suitable to deal with the arbitrary number of impression words.

Based on the above discussions, we can summarize the main technical contributions of this paper as follows:

– This paper realizes a shared latent space for shape and impression by a novel cross-modal embedding scheme. To the authors’ best knowledge, it is the first attempt to directly connect shapes (i.e., font images) and impressions by using a reliable large-scale dataset and machine-learning framework.

– Considering noisy impression words, we introduce DeepSets into the cross-modal embedding scheme. It also has another merit that we can deal with an arbitrary number of impression words for each font.

– Experimental analysis reveals the existence of two-type of impression words, shape-relevant, and shape-irrelevant. The former results in more correlated embedding in the shared latent space than the latter.

– Experimental results show that it is possible to retrieve and generate font images by specifying shape-relevant impression words.

2 Related Work

2.1 Font shape and impression

In the fields of psychology and marketing, the relationship between fonts and their impressions has been analyzed experimentally for many years [3,4,5,6,7]. These trials often use a small number of fonts. In fact, only 12 fonts are used in the rather recent trial [7]. Nowadays, analysis with larger font image datasets [1,8,9,10] has been conducted. Among them, the font-impression dataset by O’Donovan et al. [9] realizes impression-based font recommendation systems [11,12]. MyFonts dataset by Chen et al. [1] is a far more large dataset and used for impression-based font retrieval [1] and impression-specific font image generation [13].

The recent attempts are rather application-oriented and thus do not focus on the essential correlation between font styles and the impressions. To the authors’ best knowledge, this is the first attempt to understand the correlation between the shape (, or image) $X_i$ and the impression words $W_i$ of the $i$th font by embedding them into the same $d$-dimensional vector space to satisfy $f(X_i) \sim g(W_i)$ as possible, while weakening the effect of shape-irrelevant noisy impression words.
2.2 Latent space embedding

In multi-modal modeling of images and words (or texts), many attempts have been made for shared latent space embedding of the images and words. Socher et al. [14] have proposed a model that segments and annotates images by mapping images associated with the words to a latent semantic space. The same group extended this idea [15] by incorporating a neural network-based representation learning scheme of the image modality. In this work, the word modality is encoded by a hand-crafted feature, and then the image modality is mapped to the fixed word modality. In the works focusing on neural language caption generation [16,17,18], images and texts are not embedded into the same latent space explicitly, but image features by Convolutional Neural Networks (CNNs) are used as an input of Recurrent Neural Networks (RNNs) that generate textual information.

In the document analysis research field, Almazán et al. [19] have published a pioneering work that a word image and its textual information are embedded into the same space for word spotting and recognition even in a zero-shot manner. Such an embedding strategy is nowadays extended to deal with a tough multi-modal task, called Text VQA [20]. Sumi et al. [21] realized a shared latent space between online and offline handwriting sample pairs and proved that a stroke order recovery is possible via the shared latent space.

2.3 Representation learning for a set

When each training sample comprises a different number of elements without any specific order, some machine learning architecture that accepts a set as its input sample is necessary. DeepSets [2] has been proposed as a simple but powerful framework to deal with sets as samples. In recent years, Saito et al. [22] have proposed the architecture to use sets by capturing the properties from the basis of set matching mathematically and have tried novel fashion item matching using sets.

In this paper, we treat the impression words $W_i$ attached to the $i$th font as a set. The number of the attached words is different among fonts, as shown in Fig. 1. In addition, the words have no specific order. We, therefore, use DeepSets to treat $W_i$ as a set. Note that the other modality $X_i$ is represented as a stack of images instead of a set because $X_i$ always contains a fixed number of elements from ‘A’ to ‘Z.’

3 MyFonts dataset [1]

As the font dataset with impression words, we employ the dataset published by Chen et al., [1]. This dataset, hereafter called the MyFonts dataset, comprises 18,815 fonts collected at MyFonts.com. As shown in Fig. 1, each font is tagged with 0 ~ 184 impression words attached by crowd-sourcing. This means that the impression words have a large variability according to the crowd-sourcing
workers’ subjective bias. The vocabulary size of the impression words is 1,824. As noted in Section 1, impression words are often shape-relevant (such as heavy in Fig. 1) but sometimes rather shape-irrelevant (such as soccer).

Since we need to train several networks as a function $g(\cdot)$ with sufficient samples, we removed non-frequent impression words attached to less than 100 fonts. Consequently, we used 451 impression words in our experiments. In addition, we removed the dingbat (pictorial symbols) fonts and the circled fonts from the MyFonts dataset by manual inspections by three persons. We also removed fonts without any impression words (after the above non-frequent word removal). Consequently, we used 9,980, 2,992, and 1,223 fonts for training, validating, and testing, respectively. Hereafter $\Omega_{\text{train}}$, $\Omega_{\text{val}}$, and $\Omega_{\text{test}}$ denote the training, validation, and test font sets, respectively. We used 26 capital letter images of ‘A’ to ‘Z’ in each font in the later experiment since we found several fonts where small letter images are collapsed.

4 Shared Latent Space of Font Shape and Impression

This section provides the method to train our model of Fig. 3. The training is organized in a two-step manner for faster and more accurate convergence. We first perform two independent training pipelines as an initialization of the cross-modal embedding scheme. More specifically, we independently train two different autoencoders for font shapes (the bottom pipeline of Fig. 3) and impression words (the upper pipeline). The latent vectors of those autoencoders correspond to $f(X_i)$ and $g(W_i)$, respectively. Second, the end-to-end co-training will be performed to embed those latent vectors into the shared space to satisfy the condition $f(X_i) \sim g(W_i)$, while keeping the autoencoders’ outputs accurate enough.

4.1 Font shape encoding by autoencoder

An autoencoder is used for generating the latent vector $f(X_i)$ of the image modality, i.e., font shapes. As shown at the bottom of Fig. 3, the autoencoder accepts $X_i$ as its input and generate $\bar{X}_i$ via an intermediate compressed representation $f(X_i)$. Both of $X_i$ and $\bar{X}_i$ are 26 images (stacked as 26 channels) and expected to be similar with each other, i.e., $X_i \sim \bar{X}_i$, in order to guarantee that $f(X_i)$ carries the original shape information of $X_i$ sufficiently. Note that $f(X_i)$ is emitted from the autoencoder as a tensor of $7 \times 7 \times 128$, whereas it is flattened as a $d = 7 \times 7 \times 128 = 6,272$-dimensional vector in the shared latent space. In the following, these two representations are not distinguished unless otherwise mentioned ($g(W_i)$ also).

The encoder ($X_i \mapsto f(X_i)$) is based on ResNet18 (pre-trained with ImageNet) and the decoder ($f(X_i) \mapsto X_i$) is comprised of several deconvolutional

\footnote{It does not guarantee that each of the training and test sets contains more than 100 fonts for each of the 451 impression words.}
layers. (See Section 5.1 for the detail.) They are trained to minimize the construction loss function \( L_{\text{shape}} = \sum_{i=1}^{N} \| X_i - \bar{X}_i \| \), where \( N \) is the number of fonts used for training.

### 4.2 Noise-tolerant impression word encoding by DeepSets

Like the image modality, an autoencoder is used for generating the impression word vectors \( g(W_i) \). However, the word modality requires extra modules to accept an arbitrary number of impression words \( W_i = \{ w_{i,1}, \ldots, w_{i,j}, \ldots, w_{i,J_i} \} \) as its input. Moreover, each impression word \( w_{i,j} \) should be converted to a semantic vector so that similar impression words give similar effects to the system. Therefore, as shown in Fig. 3, the impression word \( w_{i,j} \) is converted to a semantic vector \( l(w_{i,j}) \) by Word2vec \[23\] (pretrained by Google News dataset), and then all the semantic vectors are aggregated to a single fixed-dimensional vector \( h(W_i) \) by DeepSets \[2\].

Fig. 4 shows how DeepSets converts the \( J_i \) semantic vectors \( \{l(w_{i,j})|j = 1, \ldots, J_i\} \) into a single vector \( h(W_i) \). DeepSets has two functions: a trainable encoding scheme, or representation learning, and an aggregation scheme. The former is a deep neural network and gives a new representation \( e(w_{i,j}) \) for the word2vec vector \( l(w_{i,j}) \). The latter is the simple averaging process \( h(W_i) = \sum_j e(w_{i,j})/J_i \). This simple aggregation scheme allows accepting any number of impression words.

The most promising property of DeepSets for our task is that it can automatically learn the feasibility of impression words for realizing the shared latent space. Therefore, if an impression word \( w_{i,j} \) is shape-irrelevant and disturbs the realization, its effect will be weakened. As the result, even though the relationship \( f(X_i) \sim g(w_{i,j}) \) will not hold for the shape-irrelevant word \( w_{i,j} \), the relationship will still hold for most shape-relevant words.

As shown at the top of Fig. 3, the autoencoder for the impression word modality accepts \( h(W_i) \) as its input and derives the latent representation \( g(W_i) \). Note that if we train the autoencoder and DeepSets in an end-to-end manner to minimize the reconstruction loss of \( h(W_i) \), it results in the trivial solution that \( h(W_i) = \bar{h}(W_i) = 0 \). We, therefore, train them to minimize the loss function \( L_{\text{impression}} = \sum_{i=1}^{N} \| s(W_i) - \bar{s}(W_i) \| \), where \( s(W_i) = \sum_j l(w_{i,j})/J_i \) (as shown in Fig. 4) and \( \bar{s}(W_i) \) is the decoder output.
4.3 Co-training for the shared latent space

After the pre-training of the autoencoder for both modalities, all the modules of both modalities are co-trained to realize the shared latent space. From its purpose to have \( f(\mathbf{X}_i) \sim g(\mathbf{W}_i) \), we have the loss function \( L_{share} = \sum_{i=1}^{N} \| f(\mathbf{X}_i) - g(\mathbf{W}_i) \| \). Consequently, the overall loss function of the proposed method becomes \( L = L_{shape} + L_{impression} + L_{share} \). In the process of minimizing the loss function \( L \), the weights of all autoencoders and DeepSets are trained simultaneously. During this, we expect that the effect of the shape-irrelevant impression words that have no clear correlation with font shapes will be minimized in DeepSets.

5 Experimental Results

5.1 Implementation details

For the image modality, the encoder \((\mathbf{X}_i \mapsto f(\mathbf{X}_i))\) is ResNet18 (pre-trained by ImageNet) that have additional convolution layer at the last whose kernel size is 1 × 1 and number of channel is 128. The decoder \((f(\mathbf{X}_i) \mapsto \mathbf{X}_i)\) is \( D^{1 \times 1}_{(512,1,0)} \) – R – \( D^{4 \times 4}_{(256,2,1)} \) – R – \( D^{4 \times 4}_{(128,2,1)} \) – R – \( D^{4 \times 4}_{(64,2,1)} \) – R – \( D^{4 \times 4}_{(32,2,1)} \) – R – \( D^{4 \times 4}_{(16,2,1)} \) where D and R show a deconvolution layer and a ReLU function respectively. The parenthesized description shows (channels, stride, padding) and the superscript shows the kernel size. For the impression word modality, the encoder \((h(\mathbf{W}_i) \mapsto g(\mathbf{W}_i))\) is \( F^{1024} \) – R – \( F^{2048} \) – R – \( F^{6272} \), and the decoder \((g(\mathbf{W}_i) \mapsto \tilde{s}(\mathbf{W}_i))\) is \( F^{2048} \) – R – \( F^{1024} \) – R – \( F^{300} \) where F shows a fully-connected layer. Note that the parenthesized description shows hidden units.

The entire network is trained by the training font set \( \Omega_{train} \) (9,980 fonts) and tested by the test set \( \Omega_{test} \) (1,223 fonts). The hyper-parameters and the training epochs are optimized by \( \Omega_{val} \) (2,992 fonts).

5.2 Quantitative analysis

We have conducted font image retrieval from a given set of impression words, which is an application task shown in Fig. [3] If a font shape \( \mathbf{X}_i \) and its corresponding impression words \( \mathbf{W}_i \) are embedded appropriately while satisfying
Table 1. Average retrieval rank, where (*) indicates a similar setup to [15].

| Method                      | $R_{\text{image} \rightarrow \text{word}}$ | $R_{\text{word} \rightarrow \text{image}}$ |
|-----------------------------|--------------------------------------------|--------------------------------------------|
| Independent                 | 608.7                                      | 612.5                                      |
| Image $\mapsto$ Word(*)     | 608.1                                      | 612.2                                      |
| Word $\mapsto$ Image(*)     | 516.9                                      | 553.0                                      |
| Proposed                    | 172.6                                      | 356.6                                      |

$f(X_i) \sim g(W_i)$, we can retrieve the font $X_i$ from $W_i$ by a simple nearest neighbor search in the shared latent space. In the following, we use a simpler setup that uses only a single impression word as the query for the retrieval. This setup allows us to understand how individual impression words are more shape-relevant or irrelevant.

Fig. 5 shows the quantitative retrieval performance on the test set $\Omega_{\text{test}}$. The performance is measured by precision at $K$ ($P@K$) at different $K$. $P@K$ indicates the ratio of the correct fonts among $K$ retrieved fonts. More specifically, we first retrieve the $K$ nearest fonts for the specified impression word $w$ by the nearest neighbor search in the latent space. Therefore, each retrieved font $X$ will satisfy $f(X) \sim g(w)$. Then, if a font $X$ has the impression word $w$ in its tag set, it is a correctly retrieved font. If $P@K$ is 1, all the $K$-neighboring font shapes have the impression word $w$ in their ground-truth.

Fig. 5 shows $P@K$ for 13 impression words which are 11 words with the highest $P@10$ and two words, retro and fun, with rather lower $P@10$ values (0 and 0.1, respectively). Most of the 11 words with higher $P@10$ are obviously shape-relevant words, such as serif, sans-serif, and bold. This proves that our framework can successfully learn the representation describing the relationship between font shapes and their impressions. It is also noteworthy that more subjective impression words such as modern and elegant, have a high $P@10$ value. Although the “elegant”ness of a font may vary among people, this result indicates there are common shape-relevant characteristics about it.

Table 1 shows a more overall evaluation result of font image retrieval performance. By giving an impression word set $W_i$ of the $i$-th test font as a query, all the images $X \in \Omega_{\text{test}}$ are then ranked by the distance $\|f(X) - g(W_i)\|$. Then, the rank $r_i$ of the correct image $X_i$ among $|\Omega_{\text{test}}|$ images is determined. Finally, the average retrieval rank $R_{\text{word} \rightarrow \text{image}} = \sum r_i / |\Omega_{\text{test}}|$ is the evaluation metric in this evaluation. In a similar manner, we can obtain the average rank $R_{\text{image} \rightarrow \text{word}}$ for the task of word retrieval with a given font image.

To our best knowledge, this is the first attempt at the cross-modal embedding of font impression and font shape into the shared latent space; therefore, there is no appropriate comparative baseline for this study. We, therefore, consider the following ablation cases to confirm the advantage of the proposed method.

- Proposed: $f(X_i)$ and $g(W_i)$ are embedded by the co-trained encoder (Section 4.3).
- Independent: $f(X_i)$ and $g(W_i)$ are embedded by the encoders by the initial training steps (Sections 4.1 and 4.2). No co-training has been made.
Fig. 6. Generating images from a single impression word and multiple impression words. Note that retro and fun are shape-irrelevant words with lower P@K, whereas the others are shape-relevant.

5.3 Font image generation with specific impressions

As shown in Fig. 3, we can use the shared latent space for generating font images with specific impressions. Feeding $g(W_i)$ to the decoder of the image modality, we can generate the alphabet images from ‘A’ to ‘Z’ in a stacked manner. This is not only an interesting application but also a test to understand how the cross modal embedding is successful for each impression word. If a generated font image for a word $w$ is not appropriate, it indicates that $w$ is a shape-irrelevant word and thus $f(X) \not\sim g(w)$.

Fig. 6(a) shows the results when generating font images with a single impression word. For the shape-relevant words such as serif to sans-serif, legible font images with the specified impression are generated successfully, thanks to
the property $f(X) \sim g(w)$. We obtained the expected results also for shape-
irrelevant words such as *fun* and *retro*. Since there is no specific trend in the
shapes for shape-irrelevant words, the generated font images are in a neutral
style. Fig. 6(b) shows the results when generating font images with multiple
impression words. The font images generated by specifying two shape-relevant
words such as (*serif*, *thin*) and (*sans-serif* and *bold*) become a mixed style suc-
sessfully. The last example shows the case that the same word is specified twice;
according to the nature of DeepSets, we can strengthen an impression by this
strategy.

We evaluated the quality of generated font images quantitatively by Haus-
dorff distance. We compared the proposed method with the Impressions2Font [13],
which is a GAN-based method for generating font images from impression words.
In this experiment, we generate font images from each impression word $w$ of a
test font $X$ by using the proposed method and Impressions2Font, respectively,
and then compare the generated image with $X$. For the comparison by Haus-
dorff distance, the generated images are binarized by the Otsu method and then
converted to edge images by the Canny method. The Hausdorff distance is cal-
culated at each of 26 alphabets and then median over them.

Fig. 7 shows the experimental results, where the horizontal axis corresponds to
the impression word $w$ sorted by the P@10 ranking for the image retrieval task.
The results demonstrate the effectiveness of the proposed method compared
with Impressions2Font, especially for impression words with high P@10 rankings.

---

4 Precisely speaking, 34 compounded impression words, such as *caps-only* and *t-shirt*,
are not acceptable by Impressions2Font and thus omitted in the evaluation. Moreover,
since we use P@10 in the evaluation of Fig. 7, we also remove the impression words
attached only to less than 10 test fonts. Consequently, we used 350 impression words
in this experiment.

5 Impressions2Font is GAN and thus can generate different images with different ran-
dom value inputs. Therefore, we used 10 random value inputs sampled from a stan-
dard normal distribution and generate $10 \times 26$ images. Consequently, we use the
median of all 260 Hausdorff distance values.

6 Since the original transitions of the Hausdorff distance values show more jagged-
ness that hides their general trends, we applied a smoothing filter to the original
transitions for getting the curves of Fig. 7.
This means the proposed method could generate font images more similar to the ground-truth images for shape-relevant impression words $w$. As the ranking of $P@K$ decreases, the distance by the proposed method gradually increases, which implies that shape-irrelevant impression words with a weak relationship between impression and shape are not embedded in the shared space. In other words, this result simply reflects the fact that it is difficult to estimate the font shape from shape-irrelevant words.

6 Conclusion

This paper showed that it is possible to realize a shared latent space where a font shape image and its multiple impression words are embedded as similar vectors. Through the shared latent space, we can handle font shapes and their impressions in a unified manner, which can lead us to generate and retrieve font images with specific fonts. Technically, we need to deal with shape-irrelevant impression words because they might disturb the unification; for this purpose, we incorporate DeepSets that can automatically weaken their effect. Experimental results revealed the existence of shape-relevant and shape-irrelevant impression words. The shape-relevant words give a higher correlation with their corresponding font shapes. The experimental results also show the possibility of impression-specific font retrieval and font generation by specifying shape-relevant impressions.

Future work will focus on additional experiments of font impression evaluation by translating font images to impression words via the shared latent space. We are also planning to standardize the impression words based on their degree of shape-relevance.

References

1. Tianlang Chen, Zhaowen Wang, Ning Xu, Hailin Jin, and Jiebo Luo. Large-scale tag-based font retrieval with generative feature learning. In ICCV, pages 9116–9125, 2019.
2. Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabás Póczos, Ruslan Salakhutdinov, and Alexander J. Smola. Deep sets. In NIPS, pages 3394–3404, 2017.
3. Terry L Childers and Jeffrey Jass. All dressed up with something to say: Effects of typeface semantic associations on brand perceptions and consumer memory. J. Consumer Psychology, 12(2):93–106, 2002.
4. John R Doyle and Paul A Bottomley. Mixed messages in brand names: Separating the impacts of letter shape from sound symbolism. Psychology & Marketing, 28(7):749–762, 2011.
5. Clive Lewis and Peter Walker. Typographic influences on reading. British J. Psychology, 80(2):241–257, 1989.
6. Dawn Shaikh and Barbara Chaparro. Perception of fonts: Perceived personality traits and appropriate uses. In Digital Fonts and Reading, chapter 13, pages 226–247. World Scientific, 2016.
7. Carlos Velasco, Andy T. Woods, Sarah Hyndman, and Charles Spence. The Taste of Typeface. i-Perception, 6(4):1–10, 2015.
8. Yuto Shinahara, Takuro Karamatsu, Daisuke Harada, Kota Yamaguchi, and Seiichi Uchida. Serif or sans: Visual font analytics on book covers and online advertisements. In *ICDAR*, pages 1041–1046, 2019.
9. Peter O’Donovan, Jānis Libeks, Aseem Agarwala, and Aaron Hertzmann. Exploratory font selection using crowdsourced attributes. *ACM Trans. Graphics*, 33(4):1–9, 2014.
10. Masaya Ikoma, Brian Kenji Iwana, and Seiichi Uchida. Effect of text color on word embeddings. In *DAS*, pages 341–355, 2020.
11. Saemi Choi, Kiyoharu Aizawa, and Nicu Sebe. Fontmatcher: Font image paring for harmonious digital graphic design. In *UII*, pages 37–41, 2018.
12. Amirreza Shirani, Franck Dernoncourt, Jose Echevarria, Paul Asente, Nedim Lipka, and Thamar Solorio. Let Me Choose: From verbal context to font selection. In *ACL*, pages 8607–8613, 2020.
13. Seiya Matsuda, Akisato Kimura, and Seiichi Uchida. Impressions2font: Generating fonts by specifying impressions. In *ICDAR*, 2021.
14. Richard Socher and Li Fei-Fei. Connecting modalities: Semi-supervised segmentation and annotation of images using unaligned text corpora. In *CVPR*, pages 966–973, 2010.
15. Richard Socher, Milind Ganjoo, Hamza Sridhar, Osbert Bastani, Christopher D. Manning, and Andrew Y. Ng. Zero-shot learning through cross-modal transfer. In *NIPS*, pages 935–943, 2013.
16. Hao Fang, Saurabh Gupta, Forrest Iandola, Rupesh K Srivastava, Li Deng, Piotr Dollár, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John C Platt, et al. From captions to visual concepts and back. In *CVPR*, pages 1473–1482, 2015.
17. Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, pages 3128–3137, 2015.
18. Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *CVPR*, pages 3156–3164, 2015.
19. J. Almazán, A. Gordo, A. Fornés, and E. Valveny. Word spotting and recognition with embedded attributes. *IEEE Trans. Patt. Anal. Mach. Intell.*, 36(12):2552–2566, 2014.
20. Ali Furkan Biten, Ruben Tito, Andres Mafla, Lluis Gomez, Marçal Rusinol, Ernest Valveny, CV Jawahar, and Dimosthenis Karatzas. Scene text visual question answering. In *ICCV*, pages 4291–4301, 2019.
21. Taichi Sumi, Brian Kenji Iwana, Hideaki Hayashi, and Seiichi Uchida. Modality conversion of handwritten patterns by cross variational autoencoders. In *ICDAR*, pages 407–412, 2019.
22. Yuki Saito, Takuma Nakamura, and Hirotaka Hachiya. Exchangeable deep neural networks for set-to-set matching and learning. In *ECCV*, pages 626–646, 2020.
23. Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119, 2013.