Illustration2Vec: A Semantic Vector Representation of Illustrations

Masaki Saito*
Tohoku University

Yusuke Matsui*
The University of Tokyo

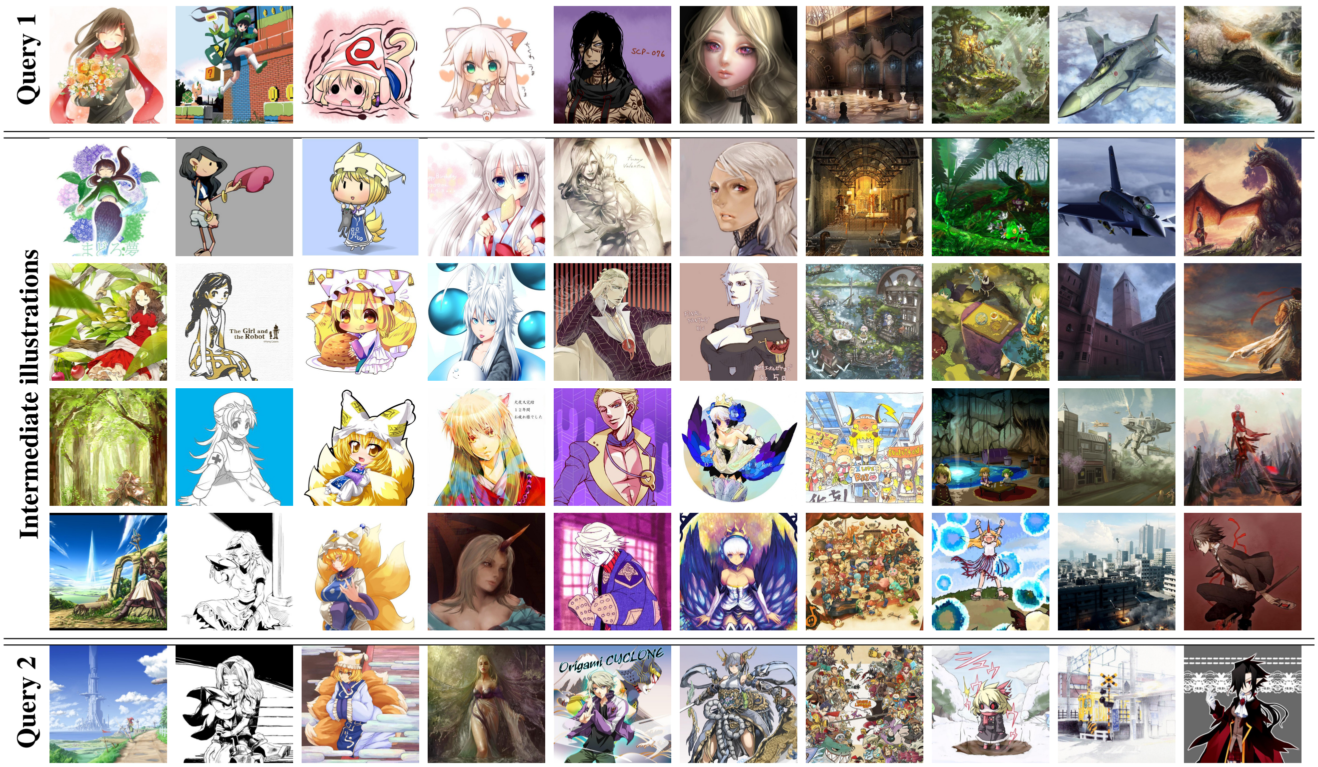

Figure 1: *Semantic morphing. In order to indicate the source, we put all the illustration links into the above figure. Note that all transitions consist of four intermediate illustrations in these examples, and we did not exclude any illustrations.*

Abstract

Referring to existing illustrations helps novice drawers to realize their ideas. To find such helpful references from a large image collection, we first build a semantic vector representation of illustrations by training convolutional neural networks. As the proposed vector space correctly reflects the semantic meanings of illustrations, users can efficiently search for references with similar attributes. Besides the search with a single query, a *semantic morphing* algorithm that searches the intermediate illustrations that gradually connect two queries is proposed. Several experiments were conducted to demonstrate the effectiveness of our methods.

CR Categories: I.4.7 [Image Processing and Computer Vision]: Feature Measurement—Feature Representation

Keywords: illustration, CNNs, visual similarity, search

1 Introduction

Drawing illustrations is a common practice for describing visual objects. It is well known that emulating or referring to existing work considerably helps drawers to depict what they really want to represent, and such reference illustrations are particularly useful for novice drawers.

With a large number of images on the web, drawers frequently find reference illustrations that would be useful for drawing. However, it is difficult for drawers to find reference illustrations from large image collections. Currently, only keyword-based or image-based searches are available to explore image collections, which are not sufficient to find references that really help in drawing. To find genuinely useful references, we need to develop techniques for analyzing, understanding, and semantically retrieving illustrations.

While thus far, many studies for analyzing natural images have been proposed, these are only a few focusing on illustrations. This lack of sufficient literature on the topic can be attributed to at least two technical issues: (1) Difficulty in recognizing illustrations. The recognition of illustrations is a more difficult problem than that of natural images because of the diversity of visual elements. For example, the size of eyes, the shape of the face, and the boldness of a pen will vary from drawer to drawer even if these drawers drew the same character. (2) Lack of a large dataset. Although recently, significant advances have been made in the field of image recognition mainly because of ImageNet (a large-scale annotated dataset),

*Authors contributed equally

no such annotated dataset exists for illustrations, thus limiting this field of study.

In this paper, we propose (1) a vector representation for illustrations, which takes the semantic difference into consideration, and (2) a novel image exploration tool for finding reference illustrations. The proposed method maps an input illustration into a 4,096-dimensional vector. The distance of two vectors represents the semantic difference between them. It is well known that feature vectors extracted from deep convolutional neural networks (CNNs) are applicable to other visual recognition tasks and are useful for a nearest neighbor search [Donahue et al. 2013]. To build a feature space representing illustrations, we train a CNN model for predicting binary attributes (also known as "tags") from a single illustration. Further, by using the proposed metric, we propose *semantic morphing*, a tool for exploring an image collection (Fig.1). By using the nearest neighbor search along with the proposed metric and semantic morphing, drawers can easily and interactively find the desired reference illustrations from an image dataset, which significantly helps them to draw their illustrations.

Our contributions are summarized as follows: (1) We propose a novel neural network model tailored for illustrations by fusing two neural network models. (2) Using the aforementioned model, we propose an exploration method for large datasets.

2 Related work

Similarity metric learning Our work is closely related to similarity metric learning methods for measuring the semantic difference of visual objects. Garces and colleagues [2014] proposed a method to learn the style of clip art from crowdsourced experiments. Such formulation was then applied to a furniture model [Liu et al. 2015] and product design [Bell and Bala 2015]. Although these researchers obtained plausible results by successfully incorporating human knowledge by crowdsourcing, collecting data still costs money and time. We do not rely on crowdsourcing and solve it directly by using a large-scale dataset.

CNN models for tag prediction CNNs have shown outstanding image classification performance in the visual recognition problem, whereas in the tag prediction problem, they may underperform compared to the conventional methods that explicitly handle part-based normalization such as poselets [Bourdev and Malik 2009]. Zhang *et al.* [2014] attributed the above results to the fact that the available training data is currently insufficient for learning pose normalization, and proposed a sophisticated model that combines both poselets and CNNs. However, this model requires an additional training set of the object parts and is inapplicable to a dataset that only includes an image and the associated tags.

3 Our approach

3.1 CNNs for attribute extraction

Dataset We first build a database consisting of approximately 1.3 million illustrations and the associated tags. We browse several web services (e.g., Danbooru and Safebooru) and collect 1,287,596 illustrations and the associated metadata. In these web services, all the tags extracted from the metadata are classified into the following four categories: *General tags* representing general attributes included in an image (e.g., "smile" and "weapon"), *copyright tags* representing the specific name of the copyright (e.g., "vocaloid"), *character tags* representing the specific name of the characters (e.g., "hatsune miku"), and *rating tags* representing X ratings ("safe," "questionable," and "explicit"). We utilize them for selecting tags. In particular, we respectively select the most frequent 512 tags from general, copyright, and character tags. Then, we concatenate these tags and three rating tags and regard 1,539 tags as a label.

VGG (model A, 11 layers)	VGG + NIN model
input ($3 \times 224 \times 224$)	
conv + max-pooling layers	
FC-4096 FC-4096 FC-1539	conv3-1024 conv3-1024 conv3-1539
sigmoid layer	average-pooling layer sigmoid layer

Table 1: *Network configurations. The layer configuration before the fully connected layers is the same as model A in [Simonyan and Zisserman 2015]. The convolutional layer parameters are denoted as "conv - (receptive field size) - (number of channels)."*

Layer configuration Modern CNNs mainly consist of two types of layers: a convolutional layer that effectively captures spatially-local correlation, and a fully connected layer (FC layer) that captures global information in an image. As most of the tags we have dealt with are associated with parts of an object (e.g., smile, glasses, etc.), we assume that in this problem local information is more significant than global information, and therefore build a new CNN model for precisely estimating tags by replacing the FC layers with the convolutional layers.

We show the configuration of our CNN in the right column of Table 1. Our network is inspired by VGG models [Simonyan and Zisserman 2015], which have achieved high performance by using very small 3×3 filters, and network-in-network (NIN) models [Lin et al. 2014], which consist of only convolutional layers and a global average pooling layer. In the configuration of convolutional layers, we refer to a VGG model with 11 layers (model A). Instead of the original VGG including 3 FC layers, we replace them with three convolutional layers and add a global average pooling layer. Unlike a multiclass classification, which is a common problem in image recognition and estimates a single discrete value, the tag prediction problem estimates multiple binary values. Therefore, we also replace the softmax layer with the sigmoid layer and train our network by minimizing the cross-entropy loss function. For the sake of comparison, we also train a standard VGG model containing FC layers (the left column of Table 1). The configuration of the network is the same as that proposed by [Simonyan and Zisserman 2015] except for replacing the 1,000-channel FC layer and the soft-max layer to the 1,539-channel FC layer and the sigmoid layer, respectively.

Using the above layer configuration and the dataset, we train the CNN from scratch. The details are shown in the supplementary material.

3.2 Binary hashing and nearest neighbor search

To generate the feature vectors that can efficiently perform a nearest neighbor search, we convert a feature vector with a real value to the binary features, and approximate the computation of the Euclidean distance to that of the Hamming distance as follows:

We first insert an additional sigmoid layer with 4,096 units after the last convolutional layer, and retrain the entire CNN initialized with the pre-trained network by using the training dataset. Then, we binarize the activation of the sigmoid layer with a threshold of 0.5 and regard it as a 4,096-bit hash string.

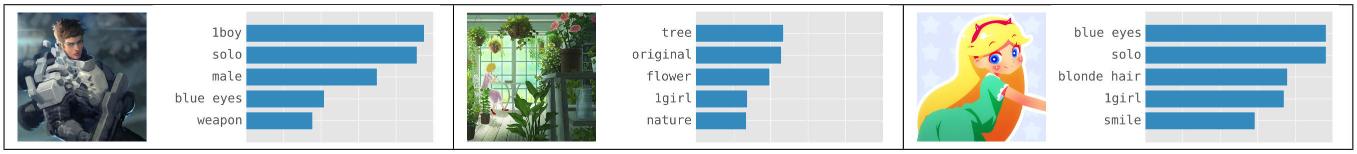

Figure 2: *Qualitative results of the tag prediction. We show the top 5 tags that have the highest confidence scores.*

	general	copyright	character	rating
Pre-trained net	10.1	4.05	6.46	59.3
VGG modelA	19.5	7.86	22.0	78.8
VGG + NIN	**32.2**	**27.8**	**57.4**	**85.9**

Table 2: *Mean average precision of four tag categories.*

4 Evaluation and Application

4.1 Tag prediction

To confirm whether our CNN correctly predicts tags, we first sent several illustrations to the CNN and examined its effectiveness. Fig.2 shows the recognition results of general tags. It is observed that our CNN precisely recognizes the gender of a character in an input image and its background such as "nature." It is notable that our network can also correctly predict the tags associated with the character such as "smile" even through the given images include a wide variety of illustration styles.

We also performed a quantitative experiment. Two models were used for the comparison. "Pre-trained net" is a simple model that uses a feature vector extracted from the last FC layer in the pre-trained VGG model (16 layers) learned from ImageNet. To predict the tags, we simply used a logistic regression and trained a binary classifier of each tag. "VGG model A" is the conventional model that we described above. In the evaluation, we used the mean average precision used in attribute classification [Zhang et al. 2014].

The results are shown in Table 2. This implies that the proposed network outperforms the conventional network across all the categories and that in the tag prediction problem, fully connected layers do not sufficiently learn pose-invariant features. We also show the average precision of each tag in Table 3.

4.2 Nearest neighbor search

We also applied the nearest neighbor search to confirm whether our network correctly finds similar images from binary features. In order to show the results in this paper, we selected 76,003 illustrations that did not contain any sexual or violent implications from the test set and used them as the database. We explicitly called this dataset, the *safe dataset*.

Fig.3 shows the results. It can be seen that for some queries, the proposed system can retrieve similar illustrations in which the same character appears. Even though the proposed system does not always retrieve illustrations containing the same character, it retrieves similar illustrations with the same attributes such as pose and illustration style.

4.3 User study for proposed metric

We performed a user study to investigate how the proposed distance metric reflects a semantic difference perceived by a human. Twenty-five participants were recruited for a user study. In the trial, we showed a participant three illustrations, namely A, B (the nearest to A in terms of the proposed metric), and C (the nearest to A in terms of the pre-trained net), then asked the participant to select the image similar to A from B or C, for each six categories (illustration style, color, composition, "from the same title or not," background, and "whether they have the same character or not"). A participant can even select a tie (neither B nor C are similar to A, or both B and C are similar to A to the same degree). We evaluated 500 trials (20 trials for each participant) by using the safe dataset.

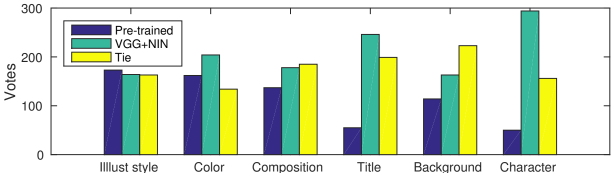

Figure 4: *A comparison of the proposed VGG + NIN model and the pre-trained net in the user study.*

The results are shown in Fig.4. The proposed VGG + NIN model collects more votes than the pre-trained model across all categories except *Illustration style* and *Background*, and particularly outperforms the pre-trained model in terms of *Title* and *Character*. Although such semantic attributes have been particularly difficult to handle, the proposed model can naturally reflect them in the metric.

4.4 Semantic morphing

Using the proposed distance metric, we created a novel tool for exploring an image collection, namely *semantic morphing*. Fig.1 shows examples. Given two query illustrations, a *semantically* smooth transition from one to another is computed (each column), by which drawers can find reference illustrations that are not reachable by traditional keyword-based search.

Given a collection of illustrations, binary features are computed; then, a distance graph is constructed where the nodes represent illustrations. The Hamming distance between two features is assigned to an edge between two corresponding nodes, where edges are created only if one node is a top 5 nearest neighbor of the other. After the distance graph is created, the smooth path between any two illustrations over the graph can be computed by finding the shortest path between the two corresponding nodes. Because the Hamming distance measures the difference in semantic attributes, the computed path over the graph shows a semantically smooth transition.

Several interesting examples of semantic morphing are shown in Fig.1. The 3rd and 4th columns from the left show smooth transitions from super-deformed to the realistic style of the similar characters. The transition of the second column from the right represents a compelling transition: battle plane, city fighting with the battle plane, and city. All of these transitions cannot be easily achieved by a keyword-based search and help drawers to come up with new ideas.

4.5 Visualization

We also visualized a set of illustrations by projecting the high-dimensional feature space onto a 2D space using t-SNE [Van der Maaten and Hinton 2008]. Fig.5 shows the result of 1,000 randomly selected illustrations from the database, where (a) to (d) show the expanded regions of the two-dimensional 2D map. As the semantic relationship of illustrations are preserved in the 2D map, we can

1girl	comic	solo	monochrome	multiple girls	4koma	blonde hair	bunnysuit	2girls	swimsuit
96.3	93.6	92.4	91.9	90.3	89.0	88.1	86.9	82.8	82.3
green hair	long hair	blue eyes	pink hair	brown hair	blue hair	red eyes	hat	black hair	green eyes
80.0	79.5	78.7	78.5	77.9	76.8	76.7	76.4	74.4	74.2
red hair	tokin hat	purple hair	one eye closed	glasses	photo	long image	third eye	purple eyes	multiple boys
72.0	70.0	68.8	68.7	68.0	67.3	66.7	65.7	65.6	65.0

Table 3: *Quantitative results of the average precision of general tags. The top 30 tags are shown. Note that in this table, we discard several inappropriate tags that contain sexual or violent connotations.*

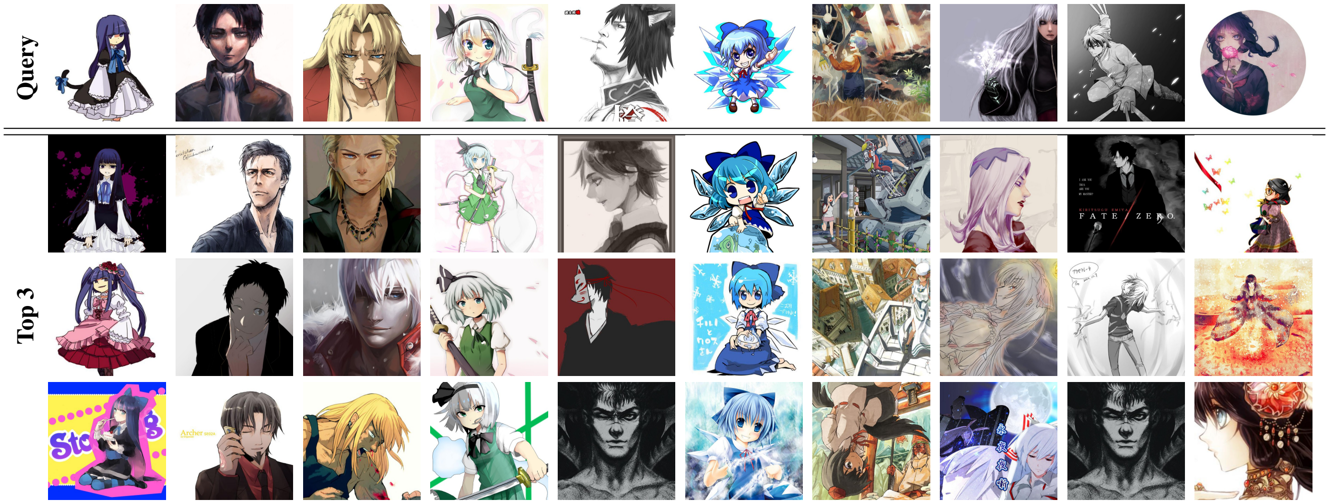

Figure 3: *Nearest neighbor retrieval: for each query, we show the top 3 illustrations from our test dataset. In order to indicate the source, we put all the illustration links into the above figure.*

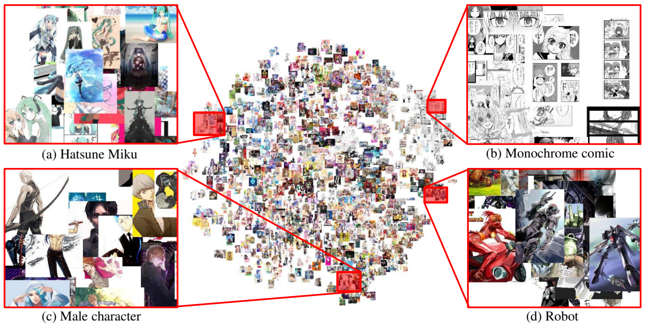
(a) Hatsune Miku
(b) Monochrome comic
(c) Male character
(d) Robot

Figure 5: *2D embedding visualization by t-SNE.*

observe several compelling image clusters. Fig.5(a) shows a cluster of *Hatsune Miku*, a green-haired girl with bunches. Images from Japanese-style four-frame comics are shown in Fig.5(b). Fig.5(c) and (d) show male characters and robot illustrations, respectively.

5 Summary

In this paper, we proposed a model for building a feature vector for illustrations. This feature is not only useful for predicting attributes in a given image but is also applicable to the nearest neighbor search. In addition, we proposed a novel method for retrieving semantically and smoothly transitioned illustrations from the two given input images. Several experiments demonstrate the effectiveness of the proposed methods.

Acknowledgments This work was supported by JSPS Grant-in-Aid for JSPS Fellows Grant Numbers 257696 and 262948. We would like to thank all the illustrators, cartoonists, and animators for motivating to begin this study. Especially, we would like to acknowledge Tonkatsu DJ Agetaro by Yujiro Koyama and Epyao for enhancing our productivity.

References

BELL, S., AND BALA, K. 2015. Learning Visual Similarity for Product Design with Convolutional Neural Networks. *ACM TOG 34*, 4.

BOURDEV, L., AND MALIK, J. 2009. Poselets: Body Part Detectors Trained Using 3D Human Pose Annotations. In *ICCV*.

DONAHUE, J., JIA, Y., VINYALS, O., HOFFMAN, J., ZHANG, N., TZENG, E., AND DARRELL, T. 2013. DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition. *CoRR abs/1310.1*.

GARCES, E., AGARWALA, A., GUTIERREZ, D., AND HERTZMANN, A. 2014. A Similarity Measure for Illustration Style. *ACM TOG 33*, 4, 93:1–93:9.

LIN, M., CHEN, Q., AND YAN, S. 2014. Network In Network. In *ICLR*.

LIU, T., HERTZMANN, A., LI, W., AND FUNKHOUSER, T. 2015. Style Compatibility for 3D Furniture Models. *ACM TOG 34*, 4.

SIMONYAN, K., AND ZISSERMAN, A. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *ICLR*.

VAN DER MAATEN, L., AND HINTON, G. 2008. Visualizing Data using t-SNE. *JMLR 9*, 85, 2579–2605.

ZHANG, N., PALURI, M., RANZATO, M., DARRELL, T., AND BOURDEV, L. 2014. PANDA: Pose Aligned Networks for Deep Attribute Modeling. In *CVPR*.