


RESEARCH

Open Access



# Feature visualization in comic artist classification using deep neural networks

Kim Young-Min\* 

\*Correspondence:  
yngmnkim@hanyang.ac.kr  
Graduate School  
of Technology & Innovation  
Management, Hanyang  
University, Wangsimni-ro,  
Seoul, South Korea

## Abstract

Deep neural networks have become a standard framework for image analytics. Besides the traditional applications, such as object classification and detection, the latest studies have started to expand the scope of the applications to include artworks. However, popular art forms, such as comics, have been ignored in this trend. This study investigates visual features for comic classification using deep neural networks. An effective input format for comic classification is first defined, and a convolutional neural network is used to classify comic images into eight different artist categories. Using a publicly available dataset, the trained model obtains a mean F1 score of 84% for the classification. A feature visualization technique is also applied to the trained classifier, to verify the internal visual characteristics that succeed in classification. The experimental result shows that the visualized features are significantly different from those of general object classification. This work represents one of the first attempts to examine the visual characteristics of comics using feature visualization, in terms of comic author classification with deep neural networks.

**Keywords:** Artistic styles, Comic classification, Convolutional neural networks, Deep neural networks, Feature visualization

## Introduction

Recent progress in computer vision has facilitated the scientific understanding of artistic visual features in artworks. Artistic style classification and style transfer are two notable examples of this type of analysis. The former aims to classify artworks into one of the predefined classes. The class type can represent the artist, genre, or painting style that effectively represents the aesthetic features of the artwork [1]. The latter aims to migrate a style from one image to another [2, 3]. This models a reference image's statistical features, which are then used to transform other images. This high-level understanding of visual features enables the effective retrieval, processing, and management of artworks. Both examples have been based on machine learning techniques in recent studies, and deep neural networks in particular. However, there is a noticeable limit in current applications, in that most existing approaches deal with fine arts. Popular art forms, such as comics, have been somewhat overlooked in this trend. Considering the present influence of popular art forms, investigating the distinguishing aspects of different types of popular artworks would be useful.

Comics is a medium expressed through juxtaposed pictorial and other images in a sequence, with the objective of delivering information or invoking an aesthetic response by viewer [4]. This is globally a very popular medium, and is currently increasing in influence thanks to the development of online comics, namely webcomics or webtoons. Despite the popularity of this medium, not many works have investigated the artistic aspects of comics in computer vision. Several aspects have been studied, such as coloring comics automatically [5] or applying style transfer to comics [6]. Anime character creation [7] and avatar creation [8] are examples of other related domains. However, these have limits in that no distinct characteristics have been examined compared to fine art.

This study attempts to tackle the problem via comic-book page classification in terms of the artistic styles expressed in the pages. A convolutional neural network (CNN), which is a standard technique in image classification, is employed as the classifier. The visual features that facilitate the classification in a trained CNN model are investigated in detail. Feature visualization is a useful tool to interpret an image classifier in ways that humans can understand. At each neuron of a trained network, a feature visualization technique is performed to reveal the neuron's visual properties. Two different input formats, comic book page and comic panel, are tested in our approach. Each image is labeled as the artist who drew the comic book.

Deep neural networks, especially convolutional neural networks have achieved a considerable success in image analysis [9, 10] and other related applications [11, 12]. ResNet [13], which have obtained the best result in the ImageNet large scale visual recognition challenge (ILSVRC) in 2015, even exceeds human recognition. While the ImageNet challenge aims to classify images into 1000 different object categories, the proposed model classifies the artwork images into fewer than 10 author categories. Therefore, a simple CNN architecture is enough for this work. Once the CNN classifier have been trained, the feature visualization technique presented in [14] is applied. In the approach, the pixels of a random noise image are updated by optimization to produce an image that can represent each neuron.

The remainder of this paper is organized as follows. In "[Related works](#)" section introduces the recent studies on image classification using deep neural networks and artwork analysis. "[Methods](#)" section deals with the proposed deep neural network structure for comic classification as well as the feature visualization using image optimization for the trained classifier. "[Results and discussion](#)" section presents the experimental results of the classification and feature visualization. Finally, the conclusions are presented in "[Conclusions](#)" section.

## **Related works**

Image classification is a representative domain of deep neural network applications. Since AlexNet [15] won the ILSVRC with a top-5 error rate of 15.4% in 2012, CNNs have become the standard frameworks for image classification. While AlexNet had only eight layers, other variations have added layers or introduced new concepts to enhance the performance. VGG-16 [16] enhanced the classification performance by increasing the layers to 16 and slightly modifying the structure. GoogLeNet [14] introduced inception modules and reduced the classification error to 6.7%. One of the most recent networks,

ResNet with “skip connections”, produced a top-5 error rate of 3.6%. The latter has 152 layers, but the new structure rather reduced the computational complexity compared to the previous models. These networks have also been successfully applied to other different kinds of recognition tasks, such as object detection and face detection.

Deep neural networks can be applied to artwork classification. Most previous studies have aimed to find effective features to represent well the paintings [17, 18]. Following the considerable success of deep learning for image classification, these techniques have been applied to the classification of art images. Firstly, CNN features have been added to the visual features describing art images and enhanced the classification accuracy [19]. Instead of CNN, a different classification method such as support vector machine had been used in the work.

Secondly, CNN classifiers have been directly applied to the art images. Various class types, such as art genre, style, and artist, have been considered. The authors of [1, 20] attempted to classify fine-art images into 27 different art style categories. They employed the WikiArt dataset with 1000 different artists, and obtained better results than previous studies using traditional classifiers. There have also been some studies dealing with other types of visual art, such as photographs [21] or illustrations [22]. These have employed CNN classifiers to identify the authorship of input images.

Meanwhile, there are very few previous studies applying deep learning techniques to the popular art forms, such as comics, until a couple of years ago. One main reason is the lack of data. Unlike fine arts, most comic books are protected by copyright. Therefore, it is difficult to construct and distribute a large-scale comic dataset. Lately, since the new comics dataset, Manga 109 was distributed in 2017, many studies in image analytics have begun to refer to this dataset. Image super-resolution is one major research area employing the dataset [23–25].

Another major area involves different analytics for comics itself. The authors of [26] introduced a new large-scale dataset of contemporary artwork including comic images. While general object recognition is applied in their work, the authors of [27] focused on the comic object detection. Four different object types have been detected in [27]. There are also studies on specialized network for comic face detection [28, 29] or comic character detection [30].

A previous study [31] revealed well a fundamental difference in comic classification from fine art classification. The work did not involve deep learning but the design of computational features from comic line segments. The authors understood well that the characteristic drawing styles of comics come from lines. This property of comics would make a difference during the training of a classifier.

With the rapid development of deep learning in visual analysis, researchers have started to interpret the trained results in ways that humans can understand. One of the attempts toward this is feature visualization, which represents each neuron of a layer in a trained neural network using the weights. Using this method, the most activated image per neuron that captures the trained characteristics of that neuron can be visualized. Feature visualization has been investigated from the primary stage of image analysis based on neural networks, and recently various additional techniques have been proposed. One main direction of current research is to find the images activating each neuron the most [32, 33]. Another is to produce an activation vector, which minimizes the

difference between a real image and its represented image from a neuron [34, 35]. This study employs the image optimization technique proposed in [14] to visualize the neurons in a trained comic classifier.

## Methods

This study consists of two main parts. First, the CNN models are trained to classify comic images into different categories, which correspond to different authors. Two input image formats are individually tested, to determine the better input image form for comic classification. The classification performance is evaluated using a publicly available comic dataset. Second, the trained models are visualized using a feature visualization technique. The two models with the different input formats are tested to examine the visual characteristics of comics in detail.

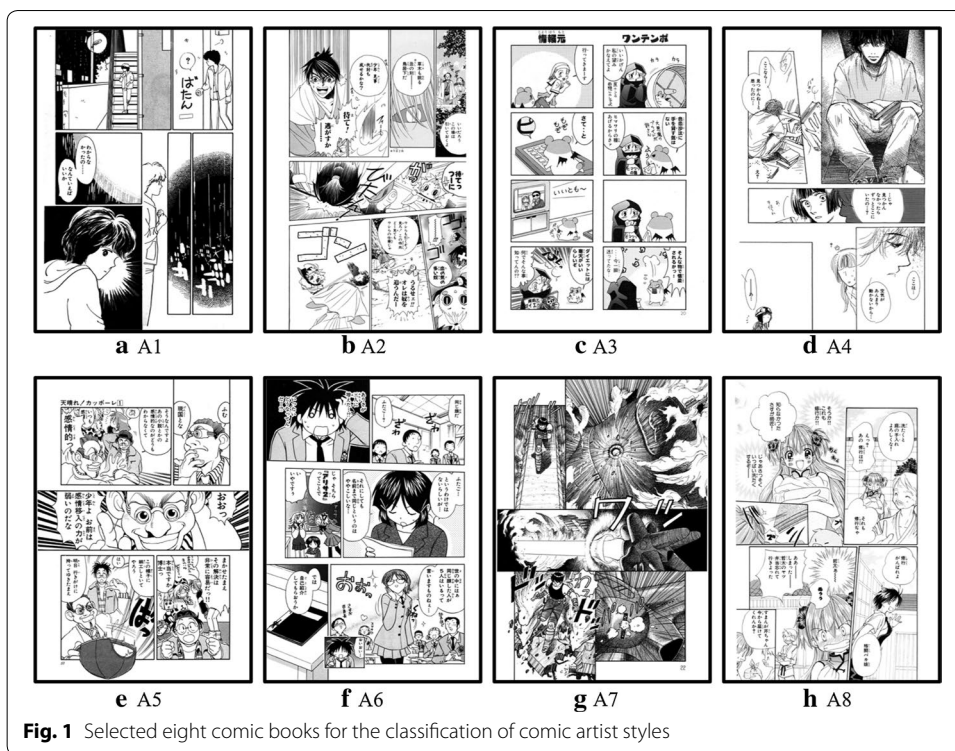
### Input data

It is first necessary to define the format of the input images to classify comic images in terms of the artistic styles. The simplest format would be the entire comic-book page. The entire page of a greyscale comic book is first employed as the input image. Each page of the comic book is scanned and filtered, to select standard pages only. A standard page is one including panels and balloons. Some unusual pages, such as those including images only, are filtered out. The second input format is the comic panel. In general, a comic page includes several panels, each of which contains a segment of action. As the drawing in a page is segmented by panels, it would be reasonable to attempt to use panels as input images. The characteristics of these two formats are compared via both classification and feature visualization.

The original data used for the experiments is the Mange 109 dataset [36], which consists of 109 manga (Japanese comic) volumes. All the volumes are drawn by different professional artists. The resolution of the scanned images is  $827 \times 1170$ . Eight volumes of the 109 are chosen for the experiments. An import supposition here is that an artist represents a distinct artistic style. Therefore, eight different comic styles are tested in our experiments. The top eight manga volumes are taken from the dataset sorted by title in ascending alphabetical order.

Figure 1 presents the examples of the selected comic pages. Each image corresponds to a representative page for each class. The ID of the artist who drew the comics is indicated at the bottom of each image. Each volume has its own distinct characteristics in drawing style. A1, A4, and A8 have the style of Shojo (girl) manga, whereas A2, A5, and A7 have a Shonen (boy) manga style. A6 is difficult to classify into one of the two types. A3 represents a special case of comics, namely four-cell manga. Table 1 shows the number of book pages used in the experiments per class.

Unlike the entire page format, the panel format needs data preprocessing to prepare input images. In other words, it is necessary to extract comic panels from the comic pages. A publicly available software is used for the extraction [37]. Some post-processing is also conducted to filter out the mis-segmented panels. Moreover, the images need to be reformatted to the same size, because the extracted panels are all different in size. Instead of adjusting resolutions, the images smaller than  $256 \times 256$  are eliminated and the larger images are cropped to  $256 \times 256$ . In the latter case, only the center part of the



**Fig. 1** Selected eight comic books for the classification of comic artist styles

**Table 1** Number of comic book pages in each class for the experiments

Artist ID	A1	A2	A3	A4	A5	A6	A7	A8	Total
#of images	181	166	126	181	169	176	149	182	1330

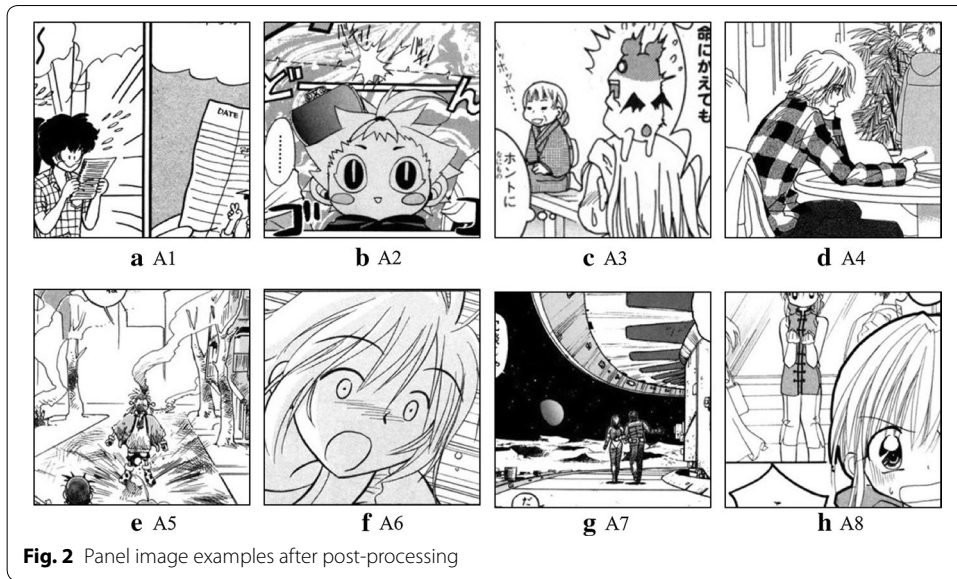
image is kept. Then, a manual post-processing removes inappropriate images for training, such as backgrounds only, parts of the body, balloons only, and images that are difficult to classify even for a human.

Figure 2 presents examples of the panel images for each class after post-processing. Even after the post-processing, there are some problematic images. For example, sample (a) contains a segmentation error and (c) includes parts of balloons with words. The examples (e) and (g) include a large background with small person area. But these types of images are kept, because it is not possible to eliminate all the problematic cases, and these cases include the distinguishable characteristics of drawing styles anyway. Table 2 shows the number of panel images in each artist class used for the experiments.

**CNN architecture for comic classification**

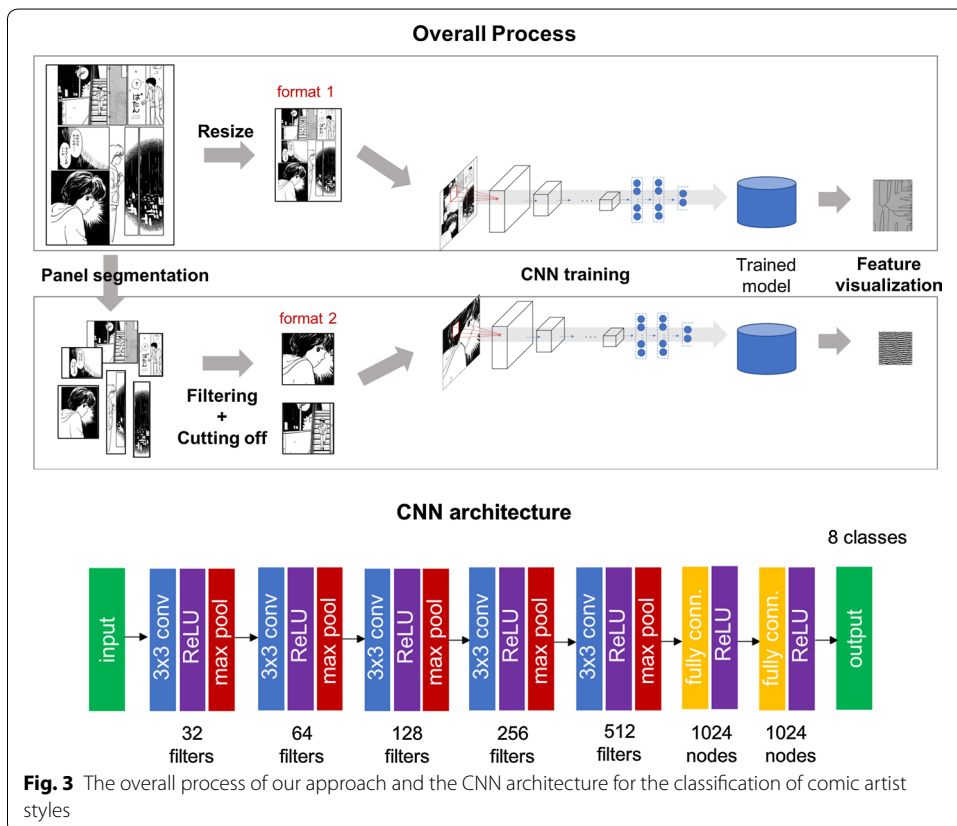
Figure 3 illustrates the overall process of the proposed approach and the detailed architecture of the CNN model. For each input image format, a CNN model is trained. And the model is used for the feature visualization in the end. As the number of classes is significantly smaller than for other major architectures, a modified version of AlexNet, one of the simplest benchmarks, is used. Filtering in the overall process means eliminating the images smaller than  $256 \times 256$  for the second format.





**Table 2** Number of panel images in each class for the experiments

Artist ID	A1	A2	A3	A4	A5	A6	A7	A8	Total
#of images	206	142	200	143	161	205	191	169	1417



The proposed network has five convolutional layers, five pooling layers, two fully connected layers, and an output layer. A ReLU activation function is applied at the end of each convolutional layer, and is followed by a max pooling. The input images consist of a greyscale image of  $300 \times 400$  pixels for the first input format and a greyscale image of  $256 \times 256$  pixels for the second. The numbers of the filters in the convolutional layers are 32, 64, 128, 256, and 512 respectively, from the first convolutional layer to the fifth. Each convolution filter has  $5 \times 5$  patches with a stride of 1. Furthermore, max pooling is employed with a  $2 \times 2$  filter and a stride of 2. Therefore, the image size is reduced by half when passing through a pooling layer. The fully connected layers have 1024 nodes each, and a ReLU function with 10% of dropout is applied.

This architecture is fixed from various pre-experiments with different settings. At each convolutional layer, different combinations of model hyperparameter values have been tested. These are the number of filters and whether pooling is applied at the end. In the architecture, the number of filters is doubled at the next convolutional layer, and max pooling is always applied, unlike AlexNet. When two convolutional layers (third and fourth) have the same number of filters without pooling between them, the performance decreases. When the number of filters at the final convolutional layer decreases, the performance also decreases. Applying repeatedly pooling layers does not degrade the final result.

### Feature visualization

Feature visualization is a useful tool for expressing the trained features of a deep neural network in image analytics. We can understand how a trained classifier can distinguish the class of an input image via feature visualization. There are many approaches, but our approach adopts a simple but powerful technique developed by the Google Brain team [14]. This involves feature visualization by image optimization and provides various regularization techniques to enhance the visualization quality. To find a representative image for a neuron, image pixels are updated via optimization by fixing the trained weights, unlike when training weights. The input image is first set to greyscale random noise. Then, the updates are repeated several times (20,000 times in our experiments), to finally obtain a feature visualization result, which represents the updated final image.

The visualization is performed for each neuron, or more specifically a channel of the trained network that corresponds to each filter in case of the convolutional layers. The objective function for the optimization at each channel can be written as follows:

$$\arg \max_I \sum_i f_i(I), \quad (1)$$

where  $I$  is the input image to be updated, and  $f_i$  is the  $i$ th activation score.

The detailed feature visualization process is represented in Algorithm 1. The visualization is conducted for a selected layer  $L$ , and a selected channel (filter)  $ch$ . We apply the forward-backward algorithm to the newly defined objective function to find the optimized image  $I^*$ . For computational efficiency, the mean value of the activation scores at the selected channel becomes the optimization objective. The function “*reduce\_mean*” computes the mean of elements across dimension of the structured channel output,

$L[:, :, :, ch]$ . The computed gradient is normalized by using the standard deviation. Finally, the image is updated using gradient ascent.

---

**Algorithm 1:** Feature visualization algorithm for a channel

---

**Data:** Random noise image  $I$   
 Selected layer and channel,  $L$  and  $ch$   
 Step size  $step$

**Result:** Updated image  $I^*$   
 $L[:, :, :, ch]$  : the output of channel  $ch$  of layer  $L$  ;  
 Define the optimization objective:  $t_{score} = reduce\_mean(L[:, :, :, ch])$ ;

**while** *not stop condition* **do**

- Forward: compute activations at  $ch$ ;
- Backward: compute gradient w.r.t. image  
 $grad \leftarrow gradients(t_{score}, I)$ ;
- Normalize gradient:  $grad \leftarrow grad / std(grad) + 1e^{-8}$ ;
- Update image:  $I \leftarrow I + grad * step$ ;

**end**

---

## Results and discussion

This section is dedicated to the experimental results for our two main contributions: comic artist style classification and feature visualization for the classifier.

### Comic artist classification

All the experiments are conducted on an NVIDIA P100 GPU. For each experiment, 80% of the images in the data were randomly selected for training, and the remainder were used for testing. The experiments were repeated 10 times by using random sub-sampling, and then the results were averaged. The total number of iterations was 30,000 with a mini-batch size of 20 for the entire-page input format, and 40,000 with the same mini-batch size for the panel input format. Under this setting, the training time was on average 90 min for entire pages and 50 min for panels.







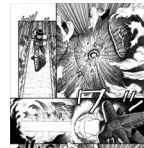

#### Entire page input format

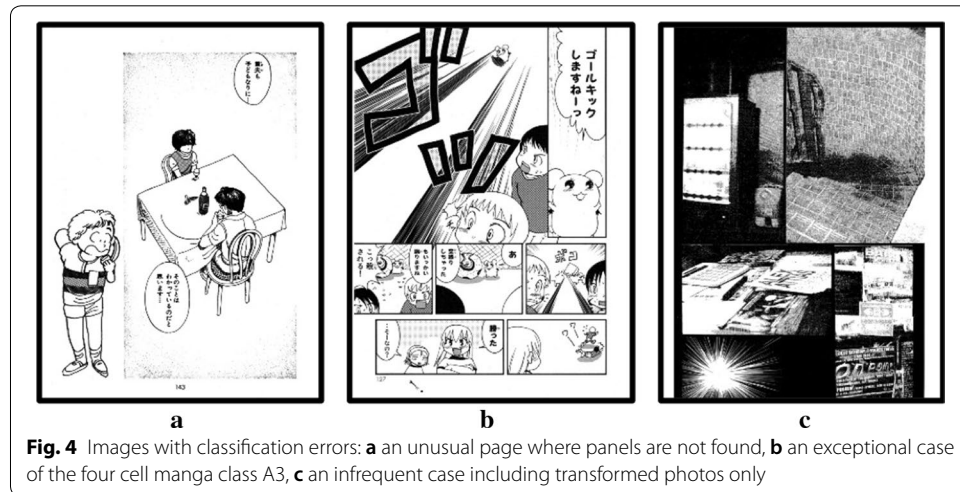
Table 3 represents the average performances of the experiments for the entire page format. The precision, recall, and F1 score are calculated for each class. All values are averaged over 10 different experiments. The total means of the averaged precision, recall, and F1 score are given in the last column of the second subtable. The mean F1 score in the experiments is 0.84. This is an encouraging result, considering that there may have been noises owing to the input format. By using the entire page, an input image includes not only drawings, but also texts, balloons, panels, and so on, which represent the different aspects of the comics.

Class A3 obtained the best performance, with an F1 score of 0.94, whereas class A2 was worst, with a score of 0.77. As A3 corresponds to four-cell manga, it is reasonable to assume that the classifier learned this special format during training. When verifying the result in detail, most of the false negatives for the class A2 were predicted as class A8, and vice versa. Considering that the difference in drawing styles between A2



**Table 3** Classification performance for the entire page format

Artist ID	A1	A2	A3	A4	
					
Precision	0.86	0.79	0.96	0.75	
Recall	0.83	0.74	0.92	0.82	
F1 score	0.85	0.77	0.94	0.79	
Artist ID	A5	A6	A7	A8	Total
					
Precision	0.84	0.88	0.82	0.80	0.84
Recall	0.76	0.84	0.87	0.86	0.83
F1 score	0.80	0.86	0.84	0.83	0.84



and A8 is smaller than for the other class pairs (see Fig. 1), the misclassification of A2 is understandable. The other classes with low F1 scores are A4 and A5, with 0.79 and 0.8, respectively. The false negatives for the class A5 were almost all predicted as A7. This explains the relatively low precision of the class A7.

Figure 4 presents three misclassified examples that correspond to the class A1, A3, and A4 respectively. The first example is an unusual page where panels are not found. The second is an exceptional case of the four cell manga class A3. This type of non-standard format was sometimes found in A3. The third example is also represents an infrequent case because it includes transformed photos only. The trained CNN model

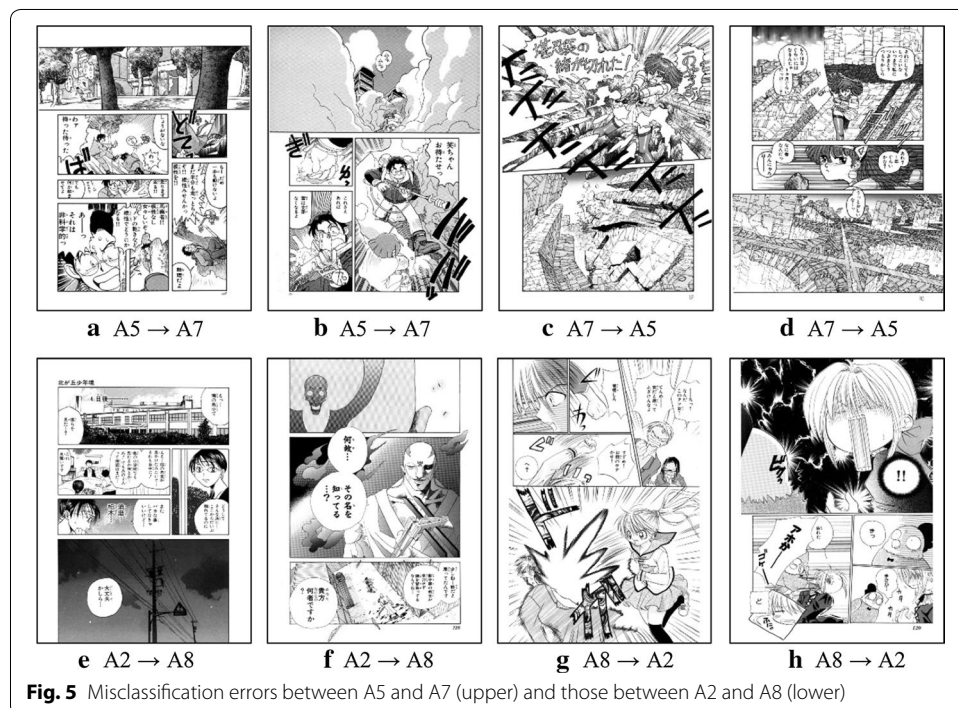
mainly misclassified these types of unusual images, of which the form had not been observed in the training set.

Besides the unusual images, there are also regular pages that are incorrectly predicted, as mentioned above. Let us examine those cases in detail. Figure 5 shows the prediction errors for two pairs of classes, A5–A7 and A2–A8. The images in the upper row show the errors between A5 and A7. The original class of the image is given before the arrow, and the predicted class is given after. Misclassification from A5 to A7 often occurs, whereas the reverse does not. The drawing styles are different from each other but both use many complicated backgrounds and effects. This common property might confuse the classifier when learning the weights. False negatives did not occur as often for class A7 as for class A5, because class A7 has a unique style, representing decorative drawing. The images in the lower row show the errors between A2 and A8. Most false negatives for the class A8 were predicted as A2. The two classes share similar drawing styles compared to the other classes, and they both employ many action lines.

Using the CNN structure, the classifier could successfully separate the images into different artist classes. There are some mistakes, but the errors are mostly because of the similarities of layouts or of drawing styles among images from different classes. This issue might be solved by adding training examples, or using an enhanced model.

**Panel input format**

Table 4 presents the average results of the experiments for the panel format. As above, all values were averaged over 10 experiments. Interestingly, the performance is considerably weaker compared to the entire page format. The mean F1 score is 0.5, and the mean precision and recall are both 0.48. Despite the weak result, class A3 again obtained the best performance. The result for A3 was impressively high, with an F1 of 0.91. This is



**Table 4** Classification performance for the panel format

Artist ID	A1	A2	A3	A4	A5	A6	A7	A8	Total
Precision	0.58	0.48	0.86	0.35	0.32	0.45	0.52	0.44	0.50
Recall	0.62	0.20	0.96	0.37	0.31	0.38	0.67	0.34	0.48
F1 score	0.60	0.29	0.91	0.36	0.32	0.41	0.59	0.38	0.48

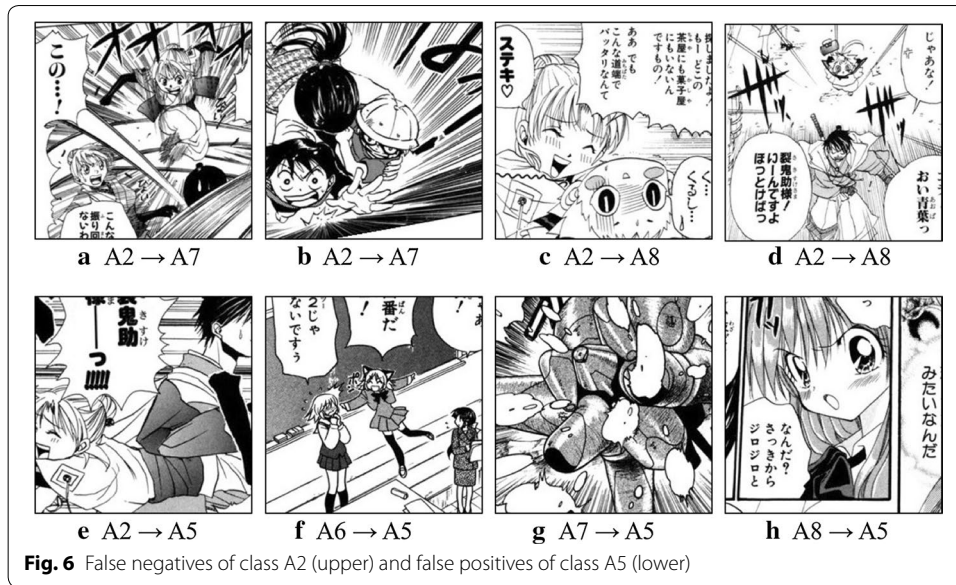
most likely because of the uniqueness of the class, which is four-cell manga. Panels in the class were clearly extracted with small errors, and in general the drawings were found in the interiors of the panels. This distinctiveness led to the exceptional score.

The classes A1 and A7 also exhibit relatively good results. These have characteristic drawing styles, where A1 prefers simple and thick lines and A7 has very decorative drawing styles with complicated patterns. On the other hand, classes A2 and A5 achieved the worst results. Their F1 scores were 0.29 and 0.32, respectively. The weak performance for A2 was mainly because of its recall of 0.20, which means that 80% of the tested images in class A2 were incorrectly predicted. Most of these were classified into the two classes, A7 and A8, and the misclassified images in the same class shared some common characteristics.

The above consequence was predictable, because in the panel format the overall layout of the page was disappeared, while the drawing styles and noises remained. Unlike paintings, the layouts of comics, such as panel structures, speech balloons, and action lines, are as important as the drawing styles. By eliminating the overall layout, the classifier should concentrate on the drawing styles and partial layout only, and therefore the training becomes more difficult. As there are not enough training data, finding patterns using mostly drawing styles becomes nontrivial.

Let us examine in detail the worst recall and precision cases, which are marked by shadow in Table 4. The upper row of Fig. 6 shows the false negative samples for the class A2 (the worst recall). As previously mentioned, their predictions were mostly A7 or A8. The examples (a) and (b) are classified as A7, whereas (c) and (d) are classified as A8. The images misclassified into A7 contain complicated action lines, whereas that into A8 include many texts. As the class A7 contains many complicated backgrounds and A8 contains more texts than the others, it is reasonable to assume that the classifier learned these properties of the classes effectively. The lower row of Fig. 6 shows the false positives for the class A5. These examples lead to the low precision for A5. Unlike the false negatives for A2 in the upper row, it is difficult to determine any pattern in the examples. As the images of A5 contain usually complicated backgrounds but the drawing lines are not very distinctive, the class is likely to share common drawing style properties with the other classes. That would be a reason for the low precision of A5. As a result, the performance gap between classes becomes wider because the classes with low performance have relatively indistinct drawing styles.

The low performance of panel format reflects the fundamental problems of the training data. There are insufficient examples, and partial layouts such as speech balloons and action lines are too often. Using the simple CNN architecture, it is difficult to extract internal patterns in the dataset.



**Feature visualization**

This subsection presents the feature visualization results for two different input formats. The visualization of neurons and the image transformation for selected neurons are provided.

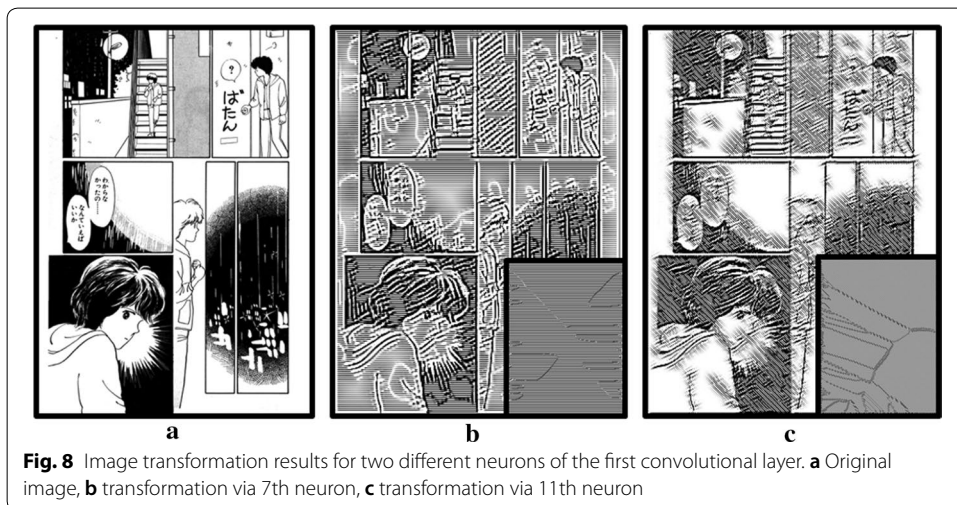
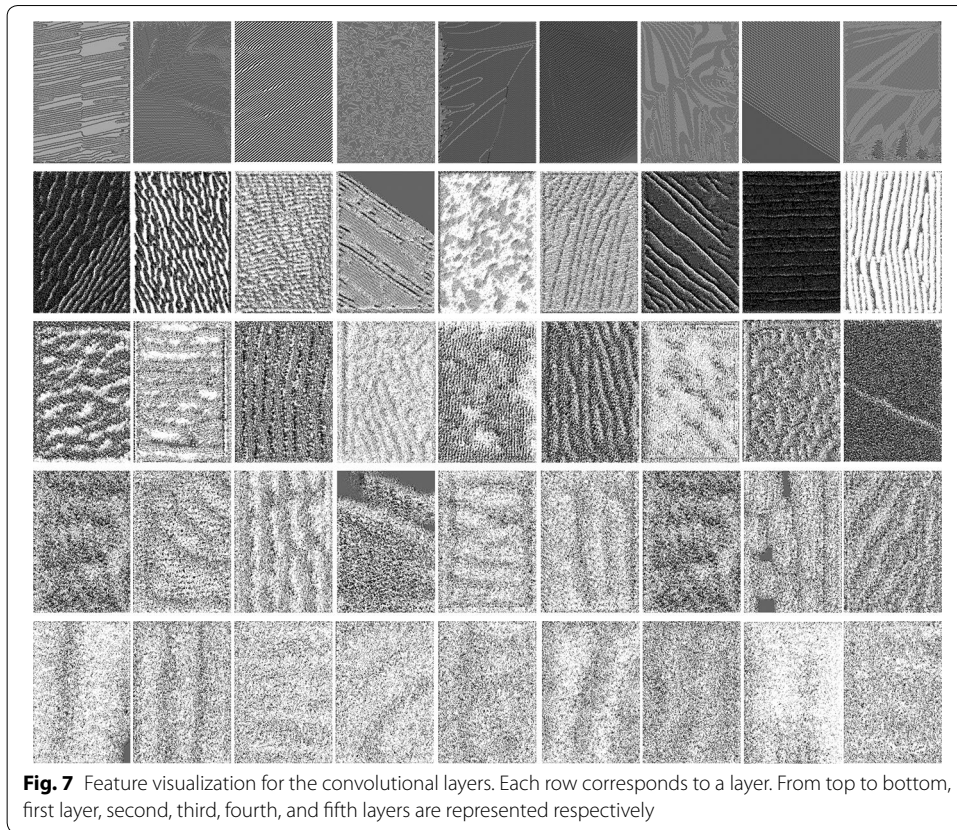
**Entire page input format**

Figure 7 presents examples of the feature visualization for the trained classifier with the entire page format. While feature visualization in object recognition networks captures each object type’s common characteristics, this is not the case for our comic classification approach. Instead of detecting object shapes, the model extracts common artistic patterns, such as textures used to separate different styles in the training set. In the figure, each row corresponds to the convolutional layer of the same number. That is, the first row represents the first convolutional layer, and so on. Nine representative neurons are selected for each layer. Some neurons do not update the input image, because the trained weights are almost zeros. There is a clear difference between the layers. The captured features in the first layer are relatively fine and dense. The extracted textures become more complicated and bolder in the upper layers. However, toward the end, the delicate patterns disappear, and only global textures remain.

Because the detected features of the comic classifier reflect the overall patterns of entire pages, the visualization cannot reflect objects. Therefore, while the general feature visualization for object classification detects more sophisticated objects in the latter layers, our classifier rather combines the textures found in the previous layers.

Besides feature visualization, image transformation for each neuron would provide an interesting option for analyzing the captured features in the neurons. Figure 8 presents the transformation results for the image using two different neurons in the first convolutional layer. Figure 8a shows the selected source image of class A1, (b) shows the transformation result with the seventh neuron, and (c) shows that of the 11th neuron. The





same technique as used for the feature visualization is employed. However, this time the input is not a random noise image, but a comic page itself. After updating the pixels of the input image 200 times, we can get the transformed result. The feature visualization of the selected neuron is shown at the bottom right of each result.

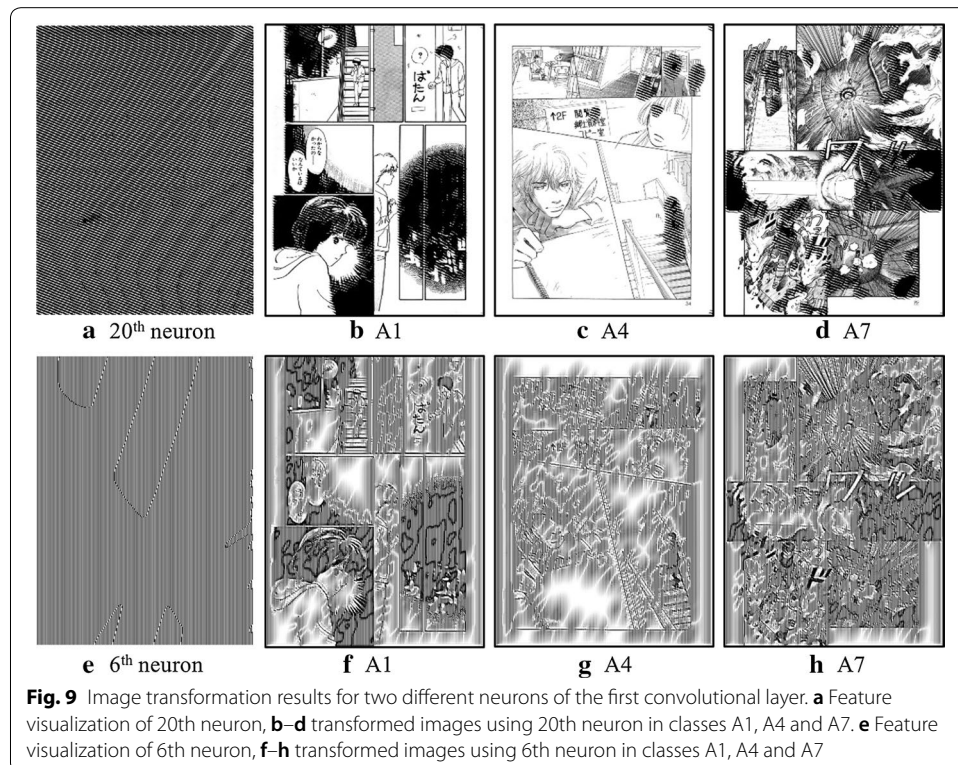
The two neurons exhibit similar feature visualization results in appearance, but the transformed images are significantly different from each other. While the seventh

neuron highlights horizontal lines, and emphasizes the outlines with white curves (b), the 11th neuron highlights diagonal lines (c). Likewise, when classifying an image with a trained model, the image is transformed by emphasizing the particular features of each neuron. Thus, at the final layer of the network, the classification is realized by aggregating these features.

To verify the image transformation more in detail, Fig. 9 illustrates the results of two other neurons (the 20th and sixth neurons). Three images from classes A1, A4, and A7, respectively, are used for the transformation. The two neurons were selected by their scores obtained when updating the test image at each neuron. A high score means that the neuron was highly activated by the image. The scores of all the neurons in the first convolutional layer are computed by updating an image. Finally, a list of scores for all neurons given an image of a certain class is obtained.

The 20th neuron was scored highly by an image of class A1 but not so highly by those of classes A4 and A7. The scores were 95, 22, and 62 for A1, A4, and A7, respectively. The feature visualization and image transformation results for the neuron are shown in the upper row of Fig. 9. An image was selected from each of the classes A1, A4, and A7. When comparing the original images with their transformations, we can discover that the dark parts of the images are emphasized during the optimization. Therefore, the image of A4, which includes tiny dark parts naturally obtained the lowest score.

Meanwhile, the sixth neuron was scored highly for all the three images. This means that the images have all been highly activated by this neuron. Thus, the captured features in the neuron would reflect the common attributes of the three images that contribute to the final classification. The transformation results are shown in the lower row of Fig. 9.





Unlike at the 20th neuron, the images were significantly modified. In the case of A4, the original drawing was nearly disappeared. This might explain the comparatively low classification performance (see Table 3).

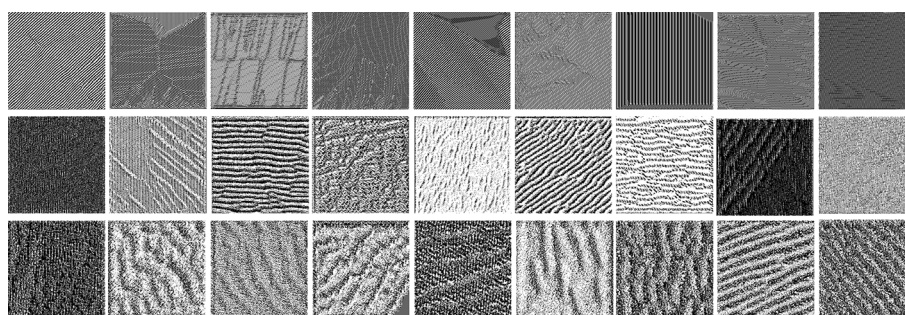
#### **Panel input format**

The feature visualization of the classifier trained using the panel input format produces almost the same result as for the entire page format, while a different visualization was expected. The reason of eliminating the panel structures was to concentrate more on the drawing styles during training. Thus, extracting more sophisticated visual features, which effectively express the drawings, was expected before training. However, the other partial layouts, such as action lines, balloons, and cuts, are still present in the panels. Moreover, as the panel also images include too various shapes in each class, the visualization could not detect representative objects for each neuron. The lack of training data could also disturb the extraction of delicate features. Thus, the layouts and drawing styles both influenced on the visualization, as in the entire page format. Some examples of the obtained visual features are represented in Fig. 10. The first three convolutional layers are shown starting from the top.

#### **Novelty in comic style feature visualization**

So far, the different aspects of feature visualization of the proposed comic classifier have been discussed. The primary difference compared to conventional object classifiers is that it does not capture objects in the neurons. The main reason is that the objective of our authorship classification approach is to categorize images in terms of drawing styles, rather than specific objects. Therefore, different objects are mixed together in a class, such that no specific shapes are detected in neurons. However, the CNN classifier could determine the internal patterns of the images in the neurons anyway. Global textures and patterns that highlight partial properties of the images have been detected via feature visualization.

There are also other distinctive characteristics of our work compared to the general image analytics. First, the target images are represented in greyscale. This makes the classification more difficult, because the color in an artwork is an important aspect of the artistic style. Second, the target images consist of drawings, or more specifically lines. Existing deep learning-based approaches dealing with paintings extract the visual



**Fig. 10** Feature visualization of the first, second, and third convolutional layers from the top. The classifier was trained with panel format

features based on textures, shapes, and patterns in two-dimensional color. On the other hand, the comics expresses textures, shapes, and patterns using lines in general. This work performed a foundational study of the visual features of the line-based artworks. For a more detailed analysis, it would be necessary to further develop specialized networks, designed to deal with those line-based artworks such as comics, drawings, and some illustrations.

Feature visualization technique used in this study had been also applied to a trained GoogLeNet [38]. Different approaches to enhance the visualization quality were proposed in that work. Diversity term, regularization, and interaction between neurons are representative examples. Although the proposed comic classifier cannot detect clear object patterns as in the work, those approaches are expected to enhance the comic feature visualization quality as well.

## Conclusions

This study proposed to use a CNN for the classification of comic styles. Comic volumes of eight artists are selected from a publicly available comic dataset for the experiments. Two different of input data formats were tested to determine the most effective format for the classification. The first was an entire-page format, and the second was a panel format. The trained model obtained an 84% mean F1 score for the former format. The experimental results are verified in detail, to demonstrate that the classifier could effectively separate the different styles, but made some errors when the styles of different classes were similar. In the case of the panel format, the trained model obtained a weak performance with an F1 score of 48%. This was mainly because of the extracted panel images, which contained too many various shapes in each class. Comparatively, distinguishing classes such as A1 and A5 achieved better results, with F1 scores over 60% and A3 obtained an exceptional score of 91%, thanks to its special layout.

The visual characteristics of a trained classifier was also investigated via a feature visualization technique. This is one of the first attempts to visualize a trained artistic style classifier. An image optimization technique was applied to the trained CNN model, to determine the visual features with which the classifier identifies the classes of test images. The visualized features were significantly different from those of general object classification. The detected features reflected the internal layouts and drawing styles of the comics, instead of representing objects.

An important drawback of our approach is that the detected features diverge strongly from the actual aesthetic elements. Although the features represent the basis of a CNN classifier effectively, they are different from the real artistic styles that distinguish artworks from a human point of view. Therefore, developing a specialized architecture, designed for the detection of aesthetic features, can be considered for future work. One of the most closely related techniques is style transfer, which transfers a style from one image to another. Combining style transfer and feature visualization for line-based artworks would represent an interesting research topic.

## Abbreviations

CNN: convolutional neural network; ILSVRC: large scale visual recognition challenge.

## Acknowledgements

Not applicable.

**Authors' contributions**

The author read and approved the final manuscript.

**Funding**

This work is partially supported by two projects, Classification of The Artists using Deep Neural Networks, funded by Hanyang University (20160000002255) and Smart Multimodal Environment of AI Chatbot Robots for Digital Healthcare (P0000536), funded by the Ministry of Trade, Industry and Energy (MOTIE).

**Availability of data and materials**

The original dataset is available on demand: <http://www.manga109.org/ja/index.html>.

**Competing interests**

The author declares that there is no competing interests.

Received: 27 April 2019 Accepted: 17 June 2019

Published online: 25 June 2019

**References**

1. Bar Y, Levy N, Wolf L. Classification of artistic styles using binarized features derived from a deep neural network. In: European conference on computer vision 2014. Springer: Cham; 2014.
2. Gatys LA, Ecker AS, Bethge M. Image style transfer using convolutional neural networks. In: 2016 IEEE conference on computer vision and pattern recognition (CVPR); 2016. p. 2414–23.
3. Chen D, Yuan L, Liao J, Yu N, Hua G. Stylebank: an explicit representation for neural image style transfer. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2017. p. 2770–9.
4. McCloud S. Understanding comics: the invisible art; 1993.
5. Hensman P, Aizawa K. cGAN-based manga colorization using a single training image. In: Proceedings of the 14th IAPR international conference on document analysis and recognition; 2017. p. 72–7.
6. Chen Y, Lai Y-K, Liu Y-J. CartoonGAN: generative adversarial networks for photo cartoonization. In: The IEEE conference on computer vision and pattern recognition (CVPR); 2018.
7. Jin Y, Zhang J, Li M, Tian Y, Zhu H, Fang Z. Towards the automatic anime characters creation with generative adversarial networks. CoRR [arxiv: abs/1708.05509](https://arxiv.org/abs/1708.05509); 2017.
8. Wolf L, Taigman Y, Polyak A. Unsupervised creation of parameterized avatars. In: IEEE international conference on computer vision (ICCV), 2017; 2017. p. 1539–47.
9. Girshick R, Donahue J, Darrell T, Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the 2014 IEEE conference on computer vision and pattern recognition, CVPR '14; 2014. p. 580–7.
10. Li H, Lin Z, Shen X, Brandt J, Hua G. A convolutional neural network cascade for face detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR); 2015. p. 5325–34.
11. Singh J, Singh G, Singh R. Optimization of sentiment analysis using machine learning classifiers. *Hum Centric Comput Inf Sci.* 2017;7(32):1–12.
12. Yuan C, Li X, Wu QMJ, Li J, Sun X. Fingerprint liveness detection from different fingerprint materials using convolutional neural network and principal component analysis. *Comput Mater Contin.* 2017;3:357–72.
13. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR); 2016. p. 770–8.
14. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A. Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR); 2015.
15. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. *Commun ACM.* 2017;60(6):84–90.
16. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. In: ICLR; 2015.
17. Johnson CR, Hendriks E, Bereznoy I, Brevedo E, Hughes S, Daubechies I, Li J, Postma E, Wang JZ. Image processing for artist identification—computerized analysis of Vincent van Gogh's painting brushstrokes. In: IEEE signal processing magazine; 2008. p. 37–48.
18. Karayev S, Trentacoste M, Han H, Agarwala A, Darrell T, Hertzmann A, Winnemoeller H. Recognizing image style. In: Proceedings of the British machine vision conference; 2014.
19. Saleh B, Elgammal AM. Large-scale classification of fine-art paintings: learning the right metric on the right feature. *Int J Digit Art Hist.* 2015:71–93.
20. Tan WR, Chan CS, Aguirre HE, Tanaka K. Ceci n'est pas une pipe: a deep convolutional network for fine-art paintings classification. In: 2016 IEEE international conference on image processing (ICIP); 2016. p. 3703–7.
21. Thomas C, Kovashka A. Seeing behind the camera: Identifying the authorship of a photograph. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2016.
22. Hicsonmez S, Samet N, Sener F, Duygulu P. Draw: deep networks for recognizing styles of artists who illustrate children's books. In: Proceedings of the 2017 ACM on international conference on multimedia retrieval. ICMR '17; 2017. p. 338–46.
23. Lai W-S, Huang J-B, Ahuja N, Yang M-H. Deep laplacian pyramid networks for fast and accurate super-resolution. In: IEEE conference on computer vision and pattern recognition; 2017.
24. Zhang Y, Tian Y, Kong Y, Zhong B, Fu Y. Residual dense network for image super-resolution. In: The IEEE conference on computer vision and pattern recognition (CVPR); 2018.
25. Haris M, Shakhnarovich G, Ukita N. Deep back-projection networks for super-resolution. In: IEEE conference on computer vision and pattern recognition (CVPR); 2018. p. 1664–73.

26. Wilber MJ, Fang C, Jin H, Hertzmann A, Collomosse J, Belongie SJ. Bam! the behance artistic media dataset for recognition beyond photography. In: IEEE international conference on computer vision (ICCV); 2017. p. 1211–20.
27. Ogawa T, Otsubo A, Narita R, Matsui Y, Yamasaki T, Aizawa K. Object detection for comics using manga109 annotations. CoRR [arxiv: abs/1803.08670](https://arxiv.org/abs/1803.08670); 2018.
28. Chu W-T, Li W-W. Manga facenet: face detection in manga based on deep neural network. In: Proceedings of the 2017 ACM on international conference on multimedia retrieval. ICMR '17; 2017. p. 412–5.
29. Nguyen N, Rigaud C, Burie J. Digital comics image indexing based on deep learning. *J Imaging*. 2018;4(7):89.
30. Nguyen N, Rigaud C, Burie J. Comic characters detection using deep learning. In: 2017 14th IAPR international conference on document analysis and recognition (ICDAR); 2017. p. 41–6.
31. Chu W-T, Chao Y-C. Line-based drawing style description for manga classification. In: Proceedings of the 22Nd ACM international conference on multimedia; 2014. p. 781–4.
32. Erhan D, Bengio Y, Courville A, Vincent P. Visualizing higher-layer features of deep networks. Technical report; 2009.
33. Yosinski J, Clune J, Nguyen AM, Fuchs TJ, Lipson H. Understanding neural networks through deep visualization. In: Proceedings of ICML—deep learning workshop; 2015.
34. Dosovitskiy A, Brox T. Inverting visual representations with convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR); 2016. p. 4829–37.
35. Mahendran A, Vedaldi A. Visualizing deep convolutional neural networks using natural pre-images. *Int J Comput Vision*. 2016;120(3):233–55.
36. Matsui Y, Ito K, Aramaki Y, Fujimoto A, Ogawa T, Yamasaki T, Aizawa K. Sketch-based manga retrieval using manga109 dataset. *Multimed Tools Appl*. 2017;76(20):21811–38.
37. Furusawa C, Hiroshiba K, Ogaki K, Odagiri Y. Comicolorization: semi-automatic manga colorization. In: SIGGRAPH Asia 2017 technical briefs; 2017. p. 12–1124.
38. Olah C, Mordvintsev A, Schubert L. Feature visualization. *Distill*. 2017. <https://doi.org/10.23915/distill.00007>.

### Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:**

- ▶ Convenient online submission
- ▶ Rigorous peer review
- ▶ Open access: articles freely available online
- ▶ High visibility within the field
- ▶ Retaining the copyright to your article

---

Submit your next manuscript at ▶ [springeropen.com](https://www.springeropen.com)

---