

FACE DETECTION AND RECOGNITION OF DRAWN CHARACTERS

HERMAN CHAU

1. INTRODUCTION

Face detection of human beings has garnered a lot of interest and research in recent years. There are quite a few relatively reliable and mature face detectors now available that perform very well, particularly on frontal face images. However, little research is done in developing technology to detect faces of drawn characters. It turns out that existing techniques such as Haar features, when applied to drawn characters is reasonably effective at detecting faces, despite the much greater variance of shapes and lack of variance in intensity across broad patches of the image. Building upon an existing face detector for drawn characters, this project explores different techniques for improving face detection and recognizing drawn characters.

By a *drawn character*, we refer to images of characters where the intensity is roughly constant in large patches of the characters. Face detection software trained on human faces perform poorly when used to detect the faces of drawn characters. In contrast, we refer to images of *realistic characters* as those in which the variation in intensity is similar to that of a photograph of a human being. Existing face detection software performs reasonably well in detecting faces of realistic characters. Figure 1 gives two examples of each to illustrate the difference between drawn characters and realistic characters:

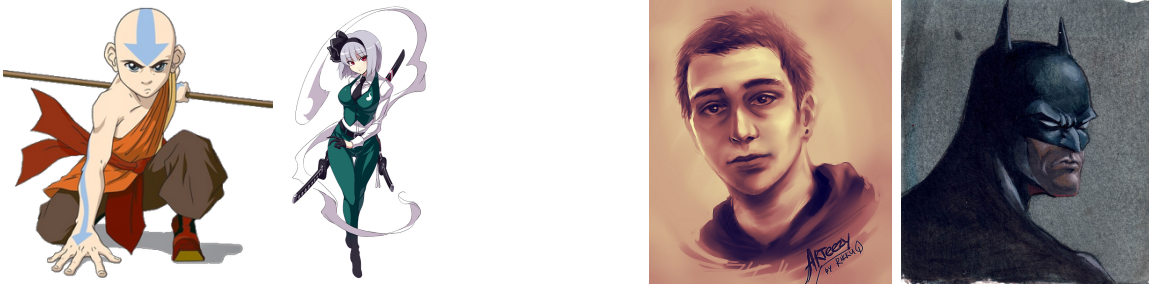


Figure 1: Drawn characters (left) and realistic characters (right)

Based on existing human face detection work, we use a Haar feature-based cascading classifier trained on a positive sample of 650 images of drawn characters and a negative sample of 200 images of scenery and compare the rate of true positives to an existing drawn character face detector. Despite our relatively small training set, we are able to achieve accuracy comparable to the existing drawn face detector.

For recognition of drawn characters, our feature extraction consists of producing a hue histogram of each image after masking out the background pixels. Most drawn characters are identifiable primarily through distinguishable clothing, hair, and eye color. Thus, we expect the hue histogram to be a feature with good predictive power. We compare three different algorithms: softmax regression, SVMs, and a k -means clustering approach.

2. EXISTING FACE DETECTION SOFTWARE

We first present a comparison two existing face detection software, using a testing set consisting of 1000 images of Japanese anime-style drawn characters. The testing set is obtained from

Safebooru, an imageboard in which various features of pictures are tagged and identified manually by volunteers. Our testing set consists of images tagged `solo`, indicating that there is a single character, and hence a single face, in the image. We test two detectors, OpenCV’s Haar Feature-based Cascading Classifier for human faces and Imager::AnimeFace, on this training set after converting each image to grayscale and performing histogram equalization. The results are summarized below in Table 2.

	Correct	False Positive
OpenCV	17.5%	2.06%
Imager::AnimeFace	53.6%	3.10%

Table 2: Accuracy Rates of OpenCV and Imager::AnimeFace on 1000 images

Beacuse OpenCV’s detector is trained on a training set consisting of human faces it fares poorly in detecting faces in drawn characters. It uses Haar Features to detect faces and a large variance in the intensity of the eye and cheek regions of the face are necessary for the detector to perform well.

The second detector we tested is Imager::AnimeFace, which is designed to detect faces in drawn characters. This performs significantly better than OpenCV’s detector, which is trained on human faces. Imager::AnimeFace uses an algorithm similar to OpenCV’s built-in cascading trainer and is trained on a dataset of roughly 70000 positive samples and 3 billion negative samples.

3. FACE DETECTION IMPROVEMENTS

We take an approach similar to Imager::AnimeFace and train a Haar Feature-based Cascading Classifier. After training on a positive training set of about 100 images, each with one face, and a negative training set of about 100 images of scenery, our classifier is able to detect faces in a testing set with limited success. Furthermore, there is an extremely high number of false positives, as depicted in Figure 3:



Figure 3: High number of false positives when using a small training set

After increasing our positive training set to 650 images, each with one face, and our negative training set to 200 images and adjusting parameters for training and detection, we are able to detect most frontal faces and even some profile faces and significantly cut down on the number of false positives. This result is depicted in Figure 4 below:

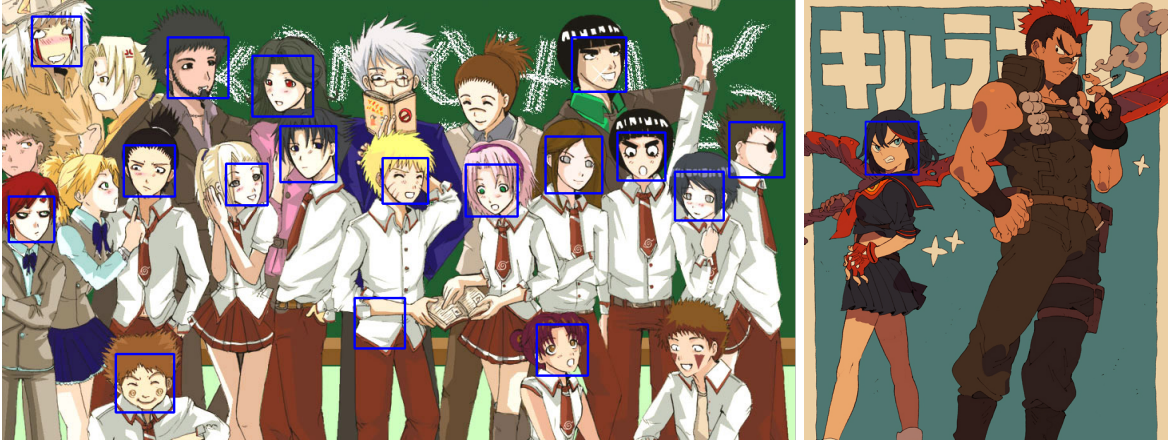


Figure 4: Significant improvements to drawn face detection

Despite the training set being roughly of the same order of magnitude, we were able to achieve much greater success in drawn face detection by appropriately tweaking parameters during training and detection. During training of the Haar feature-based cascading classifier, we increased the minimum hit rate (HR) and decreased the maximum percentage of false alarms (FA) at each stage. Although increasing HR and decreasing FA has the tendency to produce a classifier that overfits, in this case, we found that increasing HR and decreasing FA significantly improved our face detection rates. To decrease the number of false positives, we used the neighbourhood approach and increased the minimum number of neighbours required to detect a region as a face. We also made one more improvement in the form of training and detecting using grayscale images. Although the color in drawn faces are usually relatively constrained in hue and intensity, we have found that it is in fact better to use grayscale images.

We then compared the accuracy rates of `Imager::AnimeFace` and our own Haar Feature-based Cascading Classifier on detection of a different set of 1000 images. The results are summarized in Table 5 below:

	Correct	False Positive
My Cascading Classifier	51.7%	16.3%
<code>Imager::AnimeFace</code>	61.6%	5.00%

Table 5: Accuracy Rates of my cascading classifier and `Imager::AnimeFace` on 1000 images

As we can see, with the appropriate choice of parameters for training and detection, we are able to get results comparable to `Imager::AnimeFace` using a much smaller training set of positive and negative samples.

4. CHARACTER RECOGNITION THROUGH HUE HISTOGRAMS

A closely related topic to face detection is face recognition, i.e. given a picture of a drawn face, we wish to determine which character the face belongs to. In general, it is quite hard to distinguish from the faces alone, because unlike human faces, there is not much detail and features in a drawn face. Rather, humans are able to distinguish between different drawn characters primarily by the clothes they wear, their hairstyle and hair color, and other miscellaneous accessories the character wears. Thus, a relatively effective way of identifying drawn characters is based on a hue histogram of the drawing.

For a given $m \times n$ image I , we may treat it as a vector of mn integers in the interval $[0, 255]$ representing the hue of each pixel in the image. Then the hue histogram of the image I is a vector

$h \in \mathbb{R}^{256}$ where

$$h_i = \frac{\sum_{k=1}^{mn} 1\{I_{mn} = i\}}{mn}$$

That is, the i th element of h is the percentage of pixels in I with hue i . Each training example is thus a pair $(h^{(i)}, y^{(i)})$ where $h^{(i)}$ is the hue histogram of training example i and $y^{(i)} \in \{1, \dots, N\}$ is the labelling of the image. In our experimental results below, we chose $N = 10$ characters from the same series and compare the effectiveness of using softmax regression, SVMs, and k -means clustering to identify the character in a given image. In all our results, we used a training set of 30 images per character and testing set of 100 images per character.

With our feature extraction of the hue histogram from each image, applying softmax regression and SVMs is straightforward. For k -means clustering, we used N clusters on the data set with each cluster representing the “ideal” hue histogram for each character. The distance between two histograms $h^{(1)}$ and $h^{(2)}$ was defined as the ℓ_1 distance between the two vectors. A comparison of the results from each algorithm is shown in Table 6:

Class Labels	1	2	3	4	5	6	7	8	9	10	Avg
Softmax Accuracy	64%	48%	38%	41%	45%	53%	40%	39%	25%	28%	42.1%
SVM Accuracy	64%	52%	59%	46%	57%	53%	55%	47%	45%	50%	52.8%
k -means Clustering Accuracy	58%	3%	32%	33%	43%	33%	37%	13%	0%	15%	26.7%

Table 6: Accuracy Rates of Different Algorithms to Identify Characters

In all three algorithms, class 1 is identified with significantly more accuracy than the other classes. The character labelled 1 and the corresponding hue histogram obtained is shown below in Figure 7.

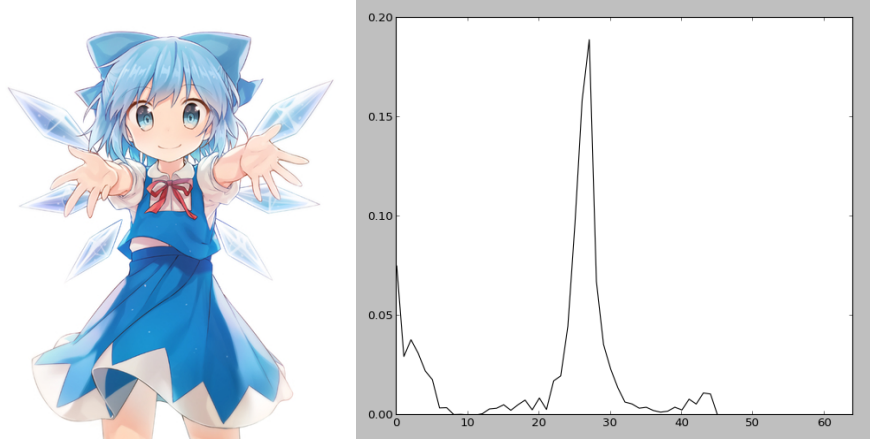


Figure 7: Example hue histogram obtained in k -means clustering

The hue histogram in this case is quite accurate, with the majority of the hues concentrated in the blue and skin color regions. Thus, the hue histogram is a reasonably good predictor of the character. Comparing the effectiveness of the algorithms, we see that SVM performs much better than both softmax regression and k -means clustering. One hypothesis to this is due to the fact that SVMs are well-equipped to deal with high-dimensional data (256 dimensional in our case here).

5. CONCLUSION AND FUTURE WORK

In our work on face detection, we have shown that we can get quite good results using a relatively small dataset by using Haar features and training with a cascading classifier. The dataset used for this paper was manually produced and hence limited in scope, but it is likely that training on

a dataset an order of magnitude larger would produce results that surpass existing face detectors such as Imager::AnimeFace. There are many other approaches to human face recognition such as Principal Component Analysis, Linear Discriminant Analysis, and Neural Networks and it would be interesting to see how such algorithms fare in detecting faces of drawn characters.

With regards to identifying drawn characters, we were able to achieve a moderate 52% success rate in identifying a character amongst 10 others. This is roughly the order of magnitude of the number of characters in a given animated series and as such, there are potential applications to automated indexing of animated films (e.g. to produce a list of scenes containing a given character). For identifying drawn characters we only extracted a hue histogram, but we would achieve much higher accuracy using if we extracted a (hue, saturation) 2D histogram. For colors of low saturation (i.e. black and white regions), the hue gives us very little information. In particular, the characters on which our classification did poorly on were characters whose color scheme was primarily black and white.

Overall, we have made some preliminary attempts in face detection and character recognition for drawn characters. As the amount of digital media continues to grow, work done in this direction has applications for the indexing of animated clips and comics.

REFERENCES

- [1] Imager::AnimeFace. <http://anime.udp.jp/face-detect/>, 2014. [Online; accessed 08-December-2014].
- [2] Kohei Takayama, Henry Johan, and Tomoyuki Nishita. Face detection and face recognition of cartoon characters using feature extraction.
- [3] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I-511. IEEE, 2001.
- [4] Xia Wan and C-C Jay Kuo. Color distribution analysis and quantization for image retrieval. In *Electronic Imaging: Science & Technology*, pages 8-16. International Society for Optics and Photonics, 1996.