Fast Genre Classification of Web Images Using Global and Local Features

Guo-Shuai Liu*, Fei Yin*, Zhen-Bo Luo‡, Cheng-Lin Liu*†
*National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences
95 Zhongguancun East Road, Beijing 100190, P.R. China
Email: {guoshuai.liu, fyin, liucl}@nlpr.ia.ac.cn
†University of Chinese Academy of Sciences, Beijing, P.R. China
‡Samsung R&D Institute China - Beijing, Machine Learning Lab
Email: zb.luo@samsung.com

***Abstract*—A number of images are present on the Web and the number is increasing every day. To effectively mine the contents embedded in Web images, it is useful to classify the images into different types so that they can be fed to different procedures for detailed analysis, such as text and non-text image discrimination. We herein propose a hierarchical algorithm for efficiently classifying Web images into four classes, namely, natural scene images, born-digital images, scanned and camera-captured paper documents, which are the most prevalent image types on the Web. Our algorithm consists of two stages; the first stage extracts global features reflecting the distributions of color, edge and gradient, and uses a support vector machine (SVM) classifier for preliminary classification. Images assigned low confidence by the first-stage classifier is processed by the second stage, which further extracts local texture features represented in the Bag-of-Words framework and uses another SVM classifier for final classification. In addition, we design two fusion strategies to train the second classifier and generate the final prediction label depending on the usage of local features in the second stage. To validate the effectiveness of our proposed method, we also build a database containing more than 55,000 images from various sources. On our test image set, we obtained an overall classification accuracy of 98.4% and the processing speed is over 27FPS on an Intel(R) Xeon(R) CPU (2.90GHz).**

***Keywords*—genre classification of Web images; low-level feature; Bag-of-Words; hierarchical classification**

I. Introduction

On the Internet and mobile network, the explosive growth of multimedia data, including texts, images and videos, brings us rich information and also the difficulty of efficiently mining relevant information. While the texts are explored by most Web mining tools, to mine the contexts in images is also very important. Particularly, the texts embedded in images provide easily understandable semantics and such images occupy a considerable proportion on Web pages. A study [1] showed that 17% of the words visible on the web pages are in image form and a large proportion (76%) of text information embedded in images cannot be found anywhere in the Web pages. The texts in images, however, are hard to extract by computers, though easily read by humans. For text detection and reading methods to process efficiently in the Internet environment, we need to quickly classify the images into different types of sources such that each type of images undertake detailed analysis by a special procedure. Also, for accurate processing, different types of text images (document images), such as natural scene text images, born-digital images, scanned and camera-captured paper documents, are better analyzed in different procedures.

In this paper, we propose a fast classification algorithm for classifying Web images into four major types, namely, natural scene images (NSI), born-digital images (BDI), scanned paper documents (SPD), and camera-captured paper documents (CPD). Natural scene images (photographs) are captured by surveillance cameras or mobile cameras, and are most popular on the Internet. Whether they contain texts or not need to be judged using more detailed procedure, but the fast identification of this image type is helpful for the overall process of Web image analysis. The other three types, BDIs, SPDs and CPDs, usually contain rich texts. They also show different characteristics of image quality, e.g., BDIs usually have large areas of constant color, and SPDs are more uniform in intensity and less distortion than CPDs. For a good tradeoff between classification accuracy and processing speed, our algorithm consists of two stages. The first stage uses global features capturing the difference of appearances between four types of images for preliminary classification with a support vector machine (SVM) classifier. Images assigned low confidence by the first-stage classifier are then processed by the second stage, which extracts local texture features encoded in the Bag-of-Words (BoW) framework and uses another SVM classifier for final classification. Compared with global features, local texture features are able to represent different patterns of color transitions and properties of edges between four types of images in a more detailed way, and yield higher classification accuracy. To validate the effectiveness of our proposed method, we also build a large image database from various sources such as Web crawling, camera capturing and other standard public databases. On our test image set, we obtained an overall classification accuracy of 98.4% and the processing speed is over 27FPS on a CPU (2.90GHz).

II. Related Work

A lot of feature extraction and classification methods were proposed in the context of content-based image retrieval (CBIR) [2], but the existing methods there are not directly

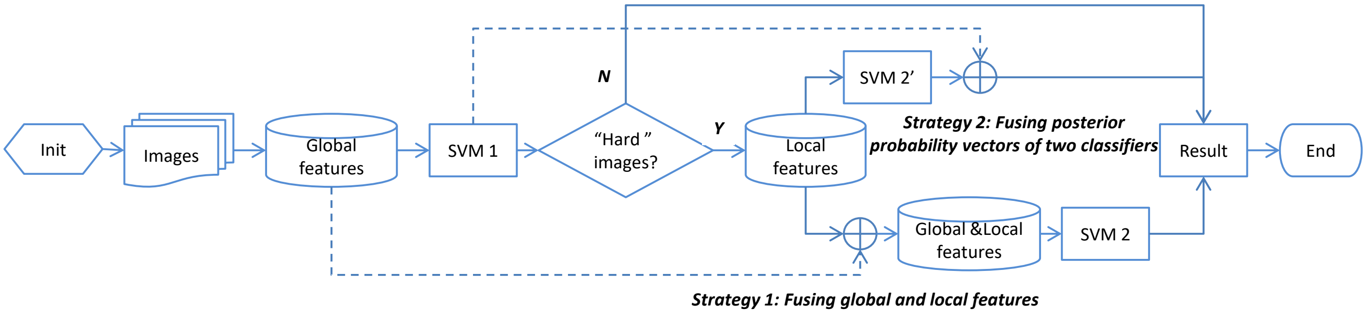

Init
Images
Global features
SVM 1
"Hard " images?
N
Y
Local features
SVM 2'
Strategy 2: Fusing posterior probability vectors of two classifiers
Result
End
Global &Local features
SVM 2
Strategy 1: Fusing global and local features

Fig. 1: A flow chart demonstrating the proposed hierarchical classification framework. If the max posterior probability from SVM 1 is higher than a given threshold T_c, the test samples will be considered "simple" and then can be made decision of corresponding class with the highest confidence. Otherwise, fusing strategies will be introduced in the second stage to find out the final result.

applicable for our purpose of image genre classification. In the following, we outline some works related to our purpose.

Hammond et al. [3] distinguished art paintings from scene photographs using color texture signatures derived from the human visual system. The receptive field profiles (RFPs) and composite visual features (CVF) they presented are helpful to solve our problem. Motivated by the physical image generation process, Ng et al. [4] proposed a novel geometry-based model for classifying photographic images and computer graphics in the context of image forgery detection. They exploited global geometry information at different scales as well as local patch statistics to discover distinctive physical characteristics of images, such as the gamma correction of photographs and the sharp structures in graphics. Despite that the method was shown effective, the feature extraction there is very time-consuming, e.g., only global fractal geometry feature extraction takes 128.1s on a 1280x1024 image. Swain et al. [5] presented a method for separating photographs and graphics on Web pages. The graphics they considered, such as corporate logos, fags, and navigation buttons, are very simple even compared to our born-digital images which contain both texts and graphics. Lienhart et al. [6] also tried to solve this problem, and the metrics they designed depend mostly on statistics of global visual cues such as color and edge orientation histogram. Lee et al. [7] tried to categorize images into art, photo and cartoon using a neural network model. Five standard MPEG-7 visual descriptors [8] in their work were employed for feature values such as Color Layout, Color Structure, Homogeneous Texture, Region Shape and Edge Histogram, which are not only redundant but also time-consuming. Pourashraf et al. [9] adopted an ensemble model for classifying images embedded in commercial real estate flyers into one of five genres, such as aerial photo, map, inside building, outside building and schematic drawing. However, the authors only evaluated their model with a small database and did not report the processing speed.

In recent years, deep neural networks, especially the convolutional neural network (CNN) [10] [11] has achieved a great success in image recognition tasks, including scene text detection and recognition [12]. The superiority of CNN is partly attributed to its ability of automatic feature exaction by learning from large training dataset. However, the CNN suffers from heavy computation in both training and testing, and so, is usually implemented using GPU for parallel computation, which to a certain extent limits its application to mobile phone and other embedded devices. It is hence not preferable for fast genre classification of images which are of huge number on the Web.

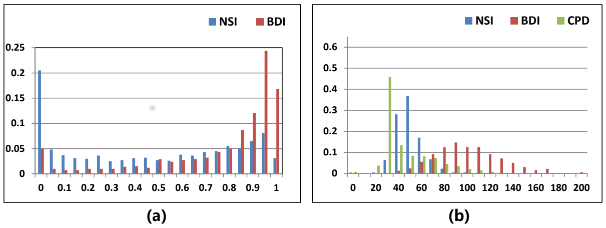
Our proposed method for fast genre classification of images uses both global visual features and local texture features which consumes low computation complexity and is of moderate dimensionality. The local texture features, extracted from different types of image patches and represented in the Bag-of-Words framework [13] [14] are shown to be effective in differentiating photographs vs. non-photo, and scanned vs. camera-captured paper documents.

III. Proposed Method

A. System Overview

Fig. 1 shows a schematic diagram of our hierarchical classification system. The first stage extracts global features and uses a SVM for preliminary classification. In this stage images with high confidence (over a threshold T_c) are made decision of class directly. Meanwhile, others with lower confidence are fed into the second-stage, which extracts local texture features represented in BoW framework and uses another SVM for final classification. In the second stage, different types of texture descriptors are extracted from local patches and each of them is represented into a BoW histogram. In particular, we carefully design four types of local patches such as edge patch, key point patch, smooth region patch and random patch. In addition, a two-step clustering method is used for a more discriminative codebook. Finally, we concatenate four BoW histograms into the local feature vector. Depending on the usage of local features, two fusion strategies are proposed to train the second classifier and generate final prediction result:

1) Training the second classifier with global and local features: Considering that global visual features alone are still



(a)

(b)

Fig. 2: The distribution of normalized histograms of different types of images over global features. (a) and (b) represent coherence of highly saturated pixels f_1 and average contrast of edge pixels f_2 respectively.

not very discriminative for those "difficult" images which are assigned to low confidence by the first classifier, one fusion strategy is to train the second classifier using global and local feature together and use its prediction label as the final result.

2) Training the second classifier with local features only: In the second way, we only use the local features to train the second classifier and then fuse the posterior probability vectors of two classifiers with different weight coefficients. Thus the test sample images will be categorized into the class that has higher confidence according to the fused vector.

B. Global Features

All the global features are designed based on the different appearances between four types of images. Compared with NSIs, BDIs tend to have fewer colors, shaper edges, larger constant color regions and more highly saturated pixels. As for the other two types, SPDs are clearly more uniform in intensity and less distortion than CPDs.

1) Coherence of highly saturated pixels f_1: This feature focuses on measuring different patterns of color transitions from pixel to pixel appearing in four types of images. NSIs often depict objects of the real world, and neither regions of uniform color nor coherent pixels of highly saturated are common in this kind of images because of the natural texture of objects, noise and diversity of illumination conditions. On the other hand, BDIs tend to have larger regions of constant color and more blocks consisting of highly saturated pixels. Let I_{rgb}, I_{hsv} and I_s denote a 3-channel RGB image, its HSV version and saturation channel respectively. A binary image I_{mask1} is obtained by thresholding I_s with a given threshold T_s. A morphological erosion operation is then performed on I_{mask1} with a 3x3 square structuring element to generate a new I_{mask2}. The number of nonzero pixels in I_{mask1} and I_{mask2} are calculated and denoted as N_1 and N_2. Finally we define $f_1 = N_2/N_1$. To demonstrate visually the effectiveness of this measure, we randomly selected 3,000 images from NSI, BDI and CPD categories in our database: 1,000 samples per class, and calculated the normalized histogram of three types of images over f_1. From Fig. 2 (a), we can observe that BDIs which have more coherent and highly saturated regions tend to have higher scores than NSIs.

2) Average contrast of edge pixels f_2: The intensity transition at edges from pixel to pixel also follows different patterns in NSIs, BDIs and CPDs. Edges in NSIs and CPDs are usually generated by occlusion, illumination and changing of surface property, while BDIs tend to have more "color edges" [3] resulting from adjacent uniform regions. Accordingly, sharp translations occur more frequently in BDIs than others. Let I_g and M_c denote a gray-scale image and its canny [15] map respectively. Firstly we define the max sharpness map M_{ms} in (1), where current pixel $p(x, y)$ and its neighbor $p(x', y')$ satisfy $max\{|x - x'|, |y - y'|\} = D$. In our experiments D is set to 2. Then f_2 can be obtained by calculating the average value of M_{ms} with M_c as the mask. As we expected, BDIs tend to have sharper edges than others in Fig. 2 (b).

$$M_{ms}(x, y) = \begin{cases} \max\{|I_g(x, y)\text{-}I_g(x', y')|\} & \text{if } M_c(x, y) > 0, \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

3) Coherence of smooth region f_3: This feature measures spatial correlation of pixels of uniform regions in images. The large flat regions in BDIs and SPDs often have high coherence and low gradient magnitude. At first, we generate horizontal and vertical gradient map M_{gx} and M_{gy} with the kernel $[-1\ 0\ 1]$ and $[-1\ 0\ 1]^T$ respectively, then an approximate gradient magnitude image M_g is obtained: $M_g = |M_{gx}| + |M_{gy}|$. Given M_g, a binary mask I_{mask3} indicating smooth regions and its eroded version I_{mask4} are generated with a threshold T_g in the same way described in Section B.1. Finally, we define $f_3 = \{N_4/N_3, N_3/N_p\}$, where N_3 and N_4 denote the number of nonzero pixels in I_{mask3} and I_{mask4} respectively, and N_p is the total number of image pixels.

4) Color histogram f_4: This feature is based on the assumption that certain colors occur more frequently in BDIs than others. Instead of directly calculating the histogram in original RGB color space, we here choose hue channel I_h for speed. The dimension of color histogram vector is set to 180.

5) Gradient magnitude histogram f_5: Last but not least, we calculated an equal interval histogram of M_g as f_5. The gradient value is changed within a range of $[0, 510]$, and the number of bins is set as 200.

C. Local Features and BoW Coding

Though successfully capturing the different characteristics of appearance in common Web images, global features proposed in the above section are still not very discriminative for classifying "difficult" images. To end this, we introduce local texture features based on the fact that local texture patterns of different types of images clearly differ from each other, e.g., BDIs and SPDs often have large constant regions, and CPDs show different texture patterns from SPDs due to the non-uniform illumination in photographing. In addition, some objects possessing certain typical texture patterns such as sky, trees or walls occur frequently in NSIs. Accordingly, we exploit four types of local patches and organize their corresponding descriptors in BoW framework [13] [14], which represents an image as a histogram of certain key descriptors and has been demonstrated very effectively in whole-image categorization tasks.

1) Local patches and descriptors: Four types of local patches are proposed in this paper, i.e., edge patch, key point patch, smooth region patch and random patch. The LBP [16] descriptors are used for the first three types of patches, and reduced color index histogram for the last. The number of each type of patch is N_{lp} and all patches have the same size: $S_{lp} \times S_{lp}$.

a) Edge patch: Inspired by the concept of "intensity edge" and "color edge" [3], we randomly select N_{lp} local patches whose centers are exactly located at canny edge point and then build an edge patch collection for each image. Combined with the BoW framework, the differences of texture in the vicinity of edge between four types of images can be introduced into the local feature and help classifying images.

b) Key point patch: In the field of object detection, interesting points on the object can be extracted to provide a "feature description" of that object. This description, extracted from training images, can then be used to identify and locate the object in another test image. Considering that certain specific objects occur frequently in particular types of images, the same technique could be used for our genre classification. We here adopt the FAST corner detection algorithm [17] to locate key points for processing speed. Similarly N_{lp} key points are randomly selected as the centers of corresponding patches.

c) Smooth region patch: Another distinctive texture comes from smooth regions, e.g., sky, lawn and water surface in NSIs, const color regions in BDIs and SPDs. Pixels coming from these regions usually have low gradient magnitude in images. Therefore we randomly select patches that have high overlap area with I_{mask3}. To make sure smooth pixels are able to occupy sufficient areas in the patches, the overlap ratio threshold is set to 0.7.

d) Random patch: As the name suggests, patches of this type are cropped randomly from original images and mostly play a complementary role to others. We use the histogram of reduced color index map of raw image to describe this type of local regions. Given the original 256^3 color space, a uniform quantization is performed and generates a 64-level (4^3) one: each axis is divided into 4 equal sized segments. We then convert the quantized 3-channel image to a 1-channel color index map by replacing the original triple value (r, g, b) with $r \times 4^2 + g \times 4^1 + b \times 4^0$ pixel by pixel. Finally a 64-D histogram based on the reduced color index map is calculated and used as random patches' descriptors.

2) Concatenated BoW representation: Since the traditional "hard" coding methods in BoW framework fails in capturing spatial layout of descriptors of local patches, we herein adopt the Locality-constrained Linear Coding (LLC) [14] algorithm to organize local descriptors. An approximate version is used for speed to incorporate locality constraint by reconstructing each descriptor with a few closest K entries in codebook. All the reconstructing vectors are then averaged to generate a final histogram vector. In order to achieve a more discriminative codebook, we also adopt a two-step clustering method; at first, for each image in training set, N_{c1} sub-centers are selected with the K-means clustering algorithm, and all the sub-centers are gathered and then clustered again to generate a codebook containing N_{c2} entries. Finally, we build codebooks for each type of patches, generate corresponding histogram vectors with LLC coding and concatenate them into a $4N_{c2}$-D vector as the final local feature.

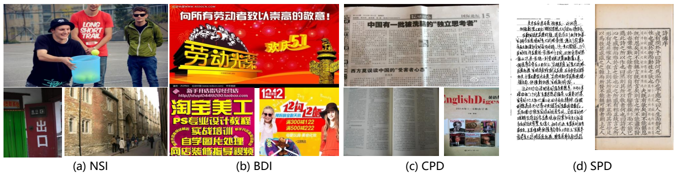

(a) NSI (b) BDI (c) CPD (d) SPD

Fig. 3: Some single-label images in SL database.

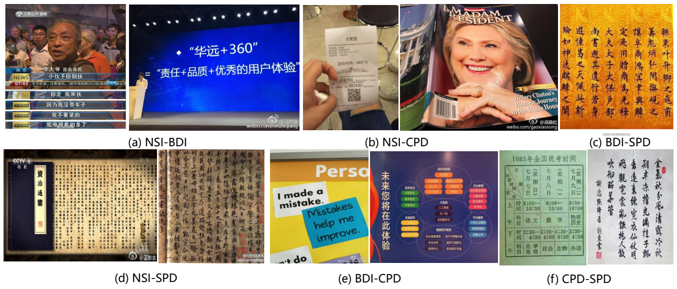

(a) NSI-BDI (b) NSI-CPD (c) BDI-SPD (d) NSI-SPD (e) BDI-CPD (f) CPD-SPD

Fig. 4: Some double-label images in DL database.

IV. Database

To validate the effectiveness of proposed method, we have built a large database of four types of images, i.e., NSI, BDI, CPD and SPD. Depending on the difficulty degree of labeling images, we divide our database into two parts: the Single-Label (SL) and the Double-Label (DL). The first one consists of such images that are easily classified by their appearances and tagged with only one label. Roughly more than 90% images in our database belong to the SL. However, there are a small fraction of "complex" samples such as photorealistic images produced by cutting edge computer graphics effects, images spliced or embedded by other different types of smaller ones, computer graphics that are displayed on LCD monitors and then recaptured by a camera, and so on. With such confusing appearances, they can not be classified intuitively into any genres. Hence, for these images, we carefully selected two proper labels as their ground truth genres in order to

TABLE I: Details of Our Image Database. There are 5,175 NSIs in our database coming from the public database SUN397, and most of the SPDs (6,036) used here come from the multilingual HW dataset.

	Single-Label	**Double-Label**				**Remarks**
		NSI	BDI	CPD	SPD	
NSI	26,410	-	-	-	-	SUN397: 5,175
BDI	12,153	1,792	-	-	-	-
CPD	6,805	1,484	282	-	-	-
SPD	6,124	12	72	51	-	MHW: 6,036
Total	51,492	3,693				55,185

TABLE II: Results on Sl Database of Proposed Features. (fps: abbreviation of “frames per second”)

Features	Acc	Speed (FPS)
Global	93.97%	28∼30
Global + Local	98.56%	16∼18

TABLE III: Results on Performance of Different Models on Sl Database. the paras number of svm mostly depends on their support vectors. (cpu(c): intel(r) xeon(r) 2.90ghz, gpu(g): nvidia titan, 12g)

Methods	Acc	Speed(FPS) (C) / (G)	Number of Paras(10^6)	Memory Usage(MB)
AlexNet (fine-tuning)	98.39%	2.46 / 157	43.02	796 (GPU)
Our(strategy 1)	98.43%	27.2 / -	5.58	-
Our(strategy 2)	98.21%	28.0 / -	3.89	-

describe them as accurately as possible. Given the complexity and variety of images on the Web, the DL of 3,693 images is a salutary supplement to SL database and makes it more proper and scalable. Totally we collected 55,185 images from various sources such as web crawling, manually camera capturing and other public databases including SUN397 [18] and the Multilingual HW Dataset [19]. More details regarding the distribution of different types of images are listed in Table I and some samples are shown in Fig. 3 and Fig. 4.

V. Experimental Results and Discussions

A. Experimental Setup

We adopt the Radial Basis Function (RBF) kernel SVM as our learning algorithm, and all the experiments were conducted with the LIBSVM [20] package. The maximum image size allowed by the system is 1000x1000 for processing speed. If the original image is larger than that, a 1000x1000 sub region will be randomly cropped and then alternatively tested. 70% images from each class are selected randomly to train classifier models, and the rest is used as testing set. Though our hierarchical algorithm involves several parameters, the ranges of their values are relatively broad. For convenience, we divide all key parameters into two parts: feature parameters and system parameters. The first one contains parameters regarding to feature extraction such as T_s, T_g, N_{lp}, S_{lp}, N_{c1} and N_{c2}. And the second consists of the confidence threshold T_c (default 0.95), and weighting coefficient w (default 0.80) of local feature classifier in the second fusion strategy, which both are critical to the overall hierarchical system. In the course of our experiments, we found out the feature parameters just slightly influence the final results. Thus we only present herein their ranges: $T_s \in [150, 230]$, $T_g \in [0.2, 2.5]$, $N_{lp} \in [100, 300]$, $S_{lp} \in [5, 30]$, $N_{c1} \in [5, 20]$ and $N_{c2} \in [50, 200]$. As for the two system parameters, we will give a detailed analysis in the following subsection.

B. Experimental Results and Discussions

1) Effectiveness of proposed features: We firstly extract global and local features from images of SL and use a SVM to train and test them directly. Results are reported in Table II. As we expected, both global and local features are discriminative for different types of Web images. Compared with using global features alone, introducing local texture information evidently increases the final classification accuracy by around 5% but at the sacrifice of processing speed.

TABLE IV: Results on Dl Database. the term “sl+dl” in parentheses means both training and testing samples come from a mixed database containing all sl and dl images. and “sl->dl” means models are trained on sl but tested on dl.

Methods	Acc (SL+DL)	Acc (SL->DL)
AlexNet (fine-tuning)	95.09%	94.13%
Our (strategy 1)	96.41%	96.11%
Our (strategy 2)	96.01%	95.33%

2) Performance of proposed hierarchical classification system: We have compared our algorithm with the popular CNN-based methods. The typical LeNet-5 [10] architecture with 128x128 input images was firstly tested on our SL database. The batch size was set to 50, and an overall accuracy of 96.7% was obtained after 40,000 iterations in training phase. For higher accuracy, we further explored the deeper AlexNet [11] architecture using a fine-tuning operation. The pre-trained CaffeNet model used here comes from the standard caffe [21] repository. In Table III, we have analyzed several performance indexes including classification accuracy, speed, number of parameters related to classifiers and GPU memory usage. Compared with direct classification (Table I), our hierarchical algorithm with two fusion strategies can achieve a comparable accuracy but at faster speed. Mainly because most “simple” samples have been confidently classified and filtered in the first stage. As for the comparison experiments, our method is evidently comparable with the AlexNet model in terms of accuracy but consumes much less computer memory. In addition, CNN is often implemented on GPUs for accelerating computation, which to a certain extent limits its application to mobile phones and other embedded devices that are usually not equipped with GPUs.

3) Robustness of different methods: We also compared our hierarchical classification system with the CNN-based model on the DL database, in which each image owns two labels. For each DL image, we randomly selected a label from its label set as the ground truth during the training phase. When testing, prediction is considered correct as long as it can meet either of both candidate ground truth labels. Results are listed in Table IV. As we can see, our system achieves a higher accuracy (SL+DL), and is more robust to the variety of different databases (SL->DL). The decline in accuracy of CNN model, we believe, is partly owing to its resizing operation, which omits the most distinctive details existed in images.

4) Effects of T_c and w on classification performance: Fig. 5 shows the effects of system parameters, and all experiments are conducted on SL database. In particular, T_c controls the number of images sent into the second stage. As the criterion becomes stringent, more and more images are selected by

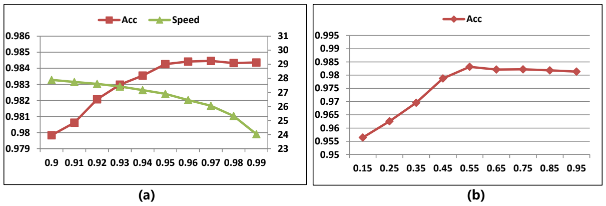


Fig. 5: Effects of system parameters on classification performance.

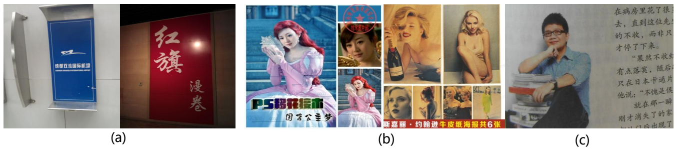


Fig. 6: Some failure cases by our classification system.

the first classifier for further analysis, which results in higher accuracy but slower speed. As the weighting coefficient of the local feature classifier in the second fusion strategy, w is critical to our hierarchical system. When it increases gradually, the final result will become more dependent on its local part. Fig. 5 (b) illustrates a range from 0.5 to 0.95 for w is expected to effectively make use of local information.

5) Analysis on misclassified images: we noticed that most failure cases come from the NSIs and BDIs. Specifically, we divide them into three categories: NSIs misclassified into BDI, BDIs misclassified into NSI, CPDs misclassified into NSI. As we can see, most misclassified NSIs in Fig. 6 (a) have large flat regions and highly saturated pixels, which violates the previous assumptions we proposed above, thus are categorized into BDI. The BDIs and CPDs containing a large proportion of scene photos in Fig. 6 (b) and (c) are more likely to be classified as NSI. In the future work, we will try to introduce more elaborated features to fix these problems.

VI. Conclusions

In this paper, we proposed a fast classification algorithm for categorizing Web images into one of four genres, i.e., natural scene images, born-digital images, camera-captured paper documents and scanned paper documents. Based on a comprehensive consideration of global and local differences of four types of images, we proposed a set of effective and efficient features derived from color, edge, gradient and texture. The hierarchical classification system we developed consists of two stages for a good tradeoff between classification accuracy and speed. We also contributed a database containing over 55,000 images. On our test image set, we obtained an overall accuracy of 98.4% and the processing speed is over 27FPS.

Our next work will be to refine the features and classifiers to improve the classification performance on images of divergent styles. To realize robust and fast reading, a task worth of attention is the fast detection of natural scene images with texts.

Acknowledgment

This work has been supported by the National Natural Science Foundation of China (NSFC) Grant 61411136002.

References

[1] A. Antonacopoulos, D. Karatzas, and J. O. Lopez, "Accessing textual information embedded in internet images," in *Proc. SPIE Internet Imaging II*, 2001, pp. 24–26.

[2] Y. Liu, D. Zhang, G. Lu, and W.-Y. Ma, "A survey of content-based image retrieval with high-level semantics," *Pattern Recognition*, vol. 40, no. 1, pp. 262–282, 2007.

[3] R. Hammoud, "Color texture signatures for art-paintings vs. scene-photographs based on human visual system," in *Proc. 17th ICPR*, vol. 2. IEEE, 2004, pp. 525–528.

[4] T.-T. Ng, S.-F. Chang, J. Hsu, L. Xie, and M.-P. Tsui, "Physics-motivated features for distinguishing photographic images and computer graphics," in *Proc. 13th ACM Multimedia*. ACM, 2005, pp. 239–248.

[5] V. Athitsos, M. J. Swain, and C. Frankel, "Distinguishing photographs and graphics on the world wide web," in *Proc. IEEE Workshop on Content-Based Access of Image and Video Libraries*. IEEE, 1997, pp. 10–17.

[6] R. Lienhart and A. Hartmann, "Classifying images on the web automatically," *Journal of Electronic Imaging*, vol. 11, no. 4, pp. 445–454, 2002.

[7] J. H. Lee, S. W. Baik, K. Kim, C. Jung, and W. Kim, "Igc: an image genre classification system," in *Proc. 3rd Artificial Intelligence and Computational Intelligence*. Springer, 2011, pp. 360–367.

[8] T. Sikora, "The mpeg-7 visual standard for content description-an overview," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 11, no. 6, pp. 696–702, 2001.

[9] P. Pourashraf, N. Tomuro, and E. Apostolova, "Genre-based image classification using ensemble learning for online flyers," in *Proc. 7th ICDIP*. ISOP, 2015, pp. 96 310Z–96 310Z.

[10] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Intelligent Signal Processing*, vol. 86, no. 11, pp. 2278–2324, 1998.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. 25th NIPS*, 2012, pp. 1097–1105.

[12] M. Jaderberg, A. Vedaldi, and A. Zisserman, "Deep features for text spotting," in *Proc. 13th ECCV*. Springer, 2014, pp. 512–528.

[13] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Proc. ECCV Workshop on Statistical Learning in Computer Vision*, vol. 1, no. 1-22. Prague, 2004, pp. 1–2.

[14] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *Proc. CVPR 2010*. IEEE, 2010, pp. 3360–3367.

[15] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, no. 6, pp. 679–698, 1986.

[16] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[17] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Proc. 9th ECCV*. Springer, 2006, pp. 430–443.

[18] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "Sun database: Large-scale scene recognition from abbey to zoo," in *Proc. CVPR 2010*. IEEE, 2010, pp. 3485–3492.

[19] "Handwritten language and writer id dataset," University of Maryland, Laboratory for Language and Media Processing (LAMP). http://lamp.cfar.umd.edu, 2016.

[20] C.-C. Chang and C.-J. Lin, "Libsvm: a library for support vector machines," *ACM Trans. Intelligent Systems and Technology*, vol. 2, no. 3, p. 27, 2011.

[21] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.